Third-order hybrid finite volume WENO method with a multilayer perceptron troubled-cell indicator for hyperbolic conservation laws

Rentian Hu[1], Chi-Wang Shu[2], Yong-Tao Zhang[3]

Abstract

Weighted essentially nonoscillatory (WENO) schemes are a popular class of numerical methods for solving hyperbolic conservation laws. Since WENO schemes are designed to deal with problems with both complicated solution structures and discontinuities / sharp gradient regions, their sophisticated nonlinear properties and high-order accuracy require more operations than many other schemes. The methodology of hybrid methods is an effective approach to decrease the computational costs and dissipation errors of WENO schemes and achieve better resolution. One of the key components for the success of hybrid WENO schemes is the application of a robust and efficient troubled-cell indicator, which detects the computational cells where the solution loses regularity. Recently, troubled-cell indicators based on artificial neural networks (ANNs) have been developed in the literature, which have the advantage of less dependence on tunable parameters and being more robust than many traditional troubled-cell indicators, and such ANN based troubled-cell indicators have been applied to hybrid finite difference WENO schemes effectively. Motivated by these works, in this paper we develop a hybrid finite volume WENO method with an ANN based troubled-cell indicator for solving hyperbolic conservation laws. While the finite difference WENO schemes are more efficient than the finite volume WENO schemes for multidimensional problems on uniform grids, the finite volume WENO schemes have the advantage such as being flexible and easy to apply on nonuniform grids. We introduce an ANN based troubled-cell indicator by constructing a multilayer perceptron (MLP) model, one of the most common ANN models. The third-order WENO scheme is focused in this paper. Extensive numerical experiments for solving various scalar equations with both convex and non-convex cases, and the Euler systems of equations on uniform and nonuniform grids of one-dimensional (1D) and two-dimensional (2D) domains, are performed to show the accuracy and nonlinear stability of the proposed hybrid finite volume WENO scheme with the MLP troubled-cell indicator. Significant accuracy improvement and computational-cost saving over the original WENO scheme are observed. Numerical experiments and comparisons with the widely-used KXRCF

[1] Department of Applied and Computational Mathematics and Statistics, University of Notre Dame, Notre Dame, IN 46556, USA. E-mail: rhu@nd.edu

[2] Division of Applied Mathematics, Brown University, Providence, RI 02912, USA. E-mail: chi-wang_shu@brown.edu. Research supported in part by NSF grant DMS-2309249.

[3] Corresponding author. Department of Applied and Computational Mathematics and Statistics, University of Notre Dame, Notre Dame, IN 46556, USA. E-mail: yzhang10@nd.edu. Research is partially supported by Simons Foundation MPS-TSM-00007854.

indicator also show the good performance of the MLP troubled-cell indicator. Although the MLP troubled-cell indicator is trained on uniform grids, it performs very well on nonuniform grids obtained by randomly perturbing uniform grids.

Key Words: Hybrid WENO methods, Finite volume methods, Multilayer perceptron, Artificial neural network, Troubled-cell indicator, Nonuniform grids, Hyperbolic conservation laws

1 Introduction

High-order accuracy numerical methods for solving hyperbolic conservation laws require careful considerations due to the possible development of discontinuities in numerical solution of the partial differential equations (PDEs) as time evolves. Without a proper treatment, spurious oscillations in numerical solution may appear, which lead to nonlinear instability of the simulation. One popular class of high-order accuracy numerical methods for solving the PDEs whose solutions may contain discontinuities and complex structures is the class of weighted essentially non-oscillatory (WENO) schemes. The first WENO scheme was constructed in [33]. In [20], the finite difference WENO schemes were developed to solve multi-dimensional problems efficiently with third-order and fifth-order accuracy, and a general framework of designing smoothness indicators and nonlinear weights for arbitrary order accuracy was provided. The main idea of the WENO schemes is to form a weighted combination of several local reconstructions based on different sub-stencils for the final WENO reconstruction. These combination coefficients are called nonlinear weights, which depend on the linear weights chosen to achieve high-order accuracy in smooth regions, and the smooth indicators that measure the smoothness of reconstructed polynomials in the relevant sub-stencils. WENO schemes have been studied extensively in the literature. For example, to deal with complex domain geometries, WENO schemes on unstructured meshes were developed in e.g. [2, 11, 17, 34, 51, 63, 64, 67]. High-order WENO schemes were applied in efficiently solving steady-state problems of hyperbolic PDEs (e.g. [57, 32, 18]), coupled with exponential integrators for solving stiff convection-diffusion-reaction PDEs (e.g. [21, 35, 58]), and implemented on sparse grids for efficiently solving high dimensional problems [68, 52, 37]. Efforts have been made to simplify the schemes or improve the accuracy and efficiency of high-order WENO schemes. Strategies include modifying the reconstruction stencils and the linear or nonlinear weights, modifying the smoothness indicators, using different approximation functions, etc., see e.g. [29, 15, 5, 67, 66, 3, 28, 19].

Since WENO methods are designed to tackle difficult problems whose solutions often have both complicated structures and discontinuities / sharp gradients, the schemes' sophisticated nonlinear properties and high-order accuracy require more computational costs than many other schemes, and they could be too dissipative for some problems. One of the effective approaches to decrease the computational costs and dissipation errors of WENO schemes and achieve better resolution is to construct a hybrid scheme. The hybrid WENO schemes combine a WENO reconstruction in stencils where the solution is non-smooth, with a high-order reconstruction of linear scheme in stencils where the solution is smooth. See e.g. [39, 8, 7, 31, 30, 55, 65]. A key component in the design of a hybrid WENO scheme is to detect the computational cells where the solution loses regularity. Such cells are often called "troubled-cells" and the techniques to identify them are referred to as "shock detectors" or "troubled-cell indicators" in the literature (see e.g. [23, 40, 41, 54]). In recent years, machine learning methods have been explored to design effective troubled-cell indicators. Troubled-cell indicators based on artificial neural networks (ANNs) have the advantage of less dependence on tunable parameters and being more robust than many traditional troubled-cell indicators. See e.g. [41, 42, 53, 61]. ANN based troubled-cell indicators have been developed and applied to hybrid finite difference WENO schemes successfully [56, 49, 59]. On the other hand, while the finite difference WENO schemes have a simple dimension-by-dimension structure and are more efficient than the finite volume WENO schemes for solving multidimensional PDEs on uniform grids [47], the finite volume WENO schemes have the advantage such as being flexible and easy to apply on arbitrary nonuniform grids.

Motivated by these aforementioned works, in this paper we develop a hybrid finite volume WENO method with an ANN based troubled-cell indicator for solving hyperbolic conservation laws on a general Cartesian mesh. The third-order finite volume WENO scheme is focused here. However, the methodology proposed in this paper is expected to be extended to higher-order finite volume WENO methods, which will be studied in our next research. A multilayer perceptron (MLP) model [43, 16], one of the most common ANN models, is constructed to obtain an ANN based troubled-cell indicator. Inspired by [49, 61], we construct the training dataset including various smooth and non-smooth functions, and numerical solutions of linear advection equations with discontinuous initial conditions which are solved by the third-order finite volume WENO scheme. This enables the MLP model to learn the feature of the numerical solutions generated by the third-order finite volume WENO scheme. The training process is performed offline on uniform grids, and the trained MLP model is added online as a troubled-cell indicator to form a hybrid finite volume WENO method. Extensive numerical experiments are performed to study the proposed hybrid scheme, which include solving various scalar equations with both convex and non-convex cases, and the Euler systems on uniform and nonuniform grids of 1D and 2D domains. Numerical results verify the desired accuracy and nonlinear stability of the developed hybrid finite volume WENO scheme with the MLP troubled-cell indicator, and significant accuracy improvement and computational-cost saving over the original WENO scheme. Furthermore, we perform extensive numerical comparisons of this MLP indicator with the widely-used KXRCF indicator [23], which show its good performance in correctly identifying troubled-cells. Numerical results also show that although the MLP troubled-cell indicator is trained on uniform grids, it performs very well on nonuniform grids obtained by randomly perturbing uniform grids.

The rest of the paper is presented as following. In Section 2, we review the third-order finite volume WENO method. In Section 3, the detailed procedure to construct the MLP model for the troubled-cell indicator is presented. The hybrid finite volume WENO method with the MLP troubled-cell indicator is described in Section 4. In Section 5, we review the KXRCF indicator which is used in the numerical comparisons. Numerical experiments are given in Section 6, and the conclusion remarks are provided in the last section.

2 Third-order finite volume WENO method

In this section, we briefly review the third-order finite volume WENO method (WENO3) for hyperbolic conservation laws. The numerical procedures on uniform meshes and nonuniform meshes are described for both 1D case and 2D case. More details can be found in e.g. [47, 33, 45].

2.1 One-dimensional case

Consider the 1D hyperbolic equation for the unknown function $u(x,t)$,

$$u_t + f(u)_x = 0, \tag{1}$$

which is defined on the spatial domain $\Omega = [a,b]$, with appropriate boundary condition and the initial condition $u(x,0) = u_0(x)$. $f(u)$ is the flux function. The spatial domain is partitioned by a mesh $a = x_{1/2} < x_{3/2} < \cdots < x_{N+1/2} = b$. Denote the computational cells as $I_i = [x_{i-1/2}, x_{i+1/2}]$, $i = 1, \cdots, N$. The center of the cell I_i is x_i, where $x_i = (x_{i-1/2} + x_{i+1/2})/2$. The mesh sizes are $\Delta x_i = x_{i+1/2} - x_{i-1/2}$, $i = 1, \cdots, N$. We integrate the equation (1) over the cell I_i to obtain

$$\frac{d\bar{u}_i(t)}{dt} = -\frac{1}{\Delta x_i}\left(f(u_{i+1/2}) - f(u_{i-1/2})\right), \qquad 1 \le i \le N. \tag{2}$$

Here $\bar{u}_i = \frac{1}{\Delta x_i}\int_{I_i} u(x,t)dx$ is the cell average of the solution $u(x,t)$ in the cell I_i. A monotone numerical flux h is used to approximate the flux f in the equation (2), e.g., the popular Lax-Friedrichs flux

$$\hat{f}_{i+1/2} = h(u^-_{i+1/2}, u^+_{i+1/2}) = \frac{1}{2}[f(u^-_{i+1/2}) + f(u^+_{i+1/2}) - \alpha(u^+_{i+1/2} - u^-_{i+1/2})], \tag{3}$$

where $\alpha = \max_u |f'(u)|$ is a constant. $u^+_{i+1/2}$ and $u^-_{i+1/2}$ are computed using the third-order WENO reconstructions based on the cell average values in the relevant stencils. With the Lax-Friedrichs flux and the WENO reconstructions, the finite volume scheme to approximate the equation (2) is

$$\frac{d\bar{u}_i(t)}{dt} = -\frac{1}{\Delta x_i}\left(\hat{f}_{i+1/2} - \hat{f}_{i-1/2}\right). \tag{4}$$

In the following we review the third-order WENO reconstruction procedure. Given the cell average values $\bar{u}_i$ for each cell I_i, $i = 1, \cdots, N$, the third-order WENO reconstructions for $u^-_{i+1/2}$ and $u^+_{i-1/2}$ are

$$\begin{aligned} u^-_{i+1/2} &= w_0 u^{(0)}_{i+1/2} + w_1 u^{(1)}_{i+1/2}, \\ u^+_{i-1/2} &= \tilde{w}_0 u^{(0)}_{i-1/2} + \tilde{w}_1 u^{(1)}_{i-1/2}. \end{aligned} \tag{5}$$

Here, we have

$$\begin{aligned} u_{i+1/2}^{(0)} &= \frac{1}{2}\bar{u}_i + \frac{1}{2}\bar{u}_{i+1}, \\ u_{i+1/2}^{(1)} &= -\frac{1}{2}\bar{u}_{i-1} + \frac{3}{2}\bar{u}_i, \\ u_{i-1/2}^{(0)} &= \frac{3}{2}\bar{u}_i - \frac{1}{2}\bar{u}_{i+1}, \\ u_{i-1/2}^{(1)} &= \frac{1}{2}\bar{u}_{i-1} + \frac{1}{2}\bar{u}_i \end{aligned} \tag{6}$$

on a uniform grid. On nonuniform grids, the coefficients in (6) are not constants anymore. They depend on the mesh sizes of the stencil and are pre-computed as in [47]. The explicit formulae to calculate $u_{i+1/2}^{(r)}$ and $u_{i-1/2}^{(r)}$, $r = 0, 1$, are

$$\begin{aligned} u_{i+1/2}^{(0)} &= \left(\frac{\Delta x_{i+1}}{\Delta x_{i+1} + \Delta x_i}\right)\bar{u}_i + \left(\frac{\Delta x_i}{\Delta x_{i+1} + \Delta x_i}\right)\bar{u}_{i+1}, \\ u_{i+1/2}^{(1)} &= \left(-\frac{\Delta x_i}{\Delta x_i + \Delta x_{i-1}}\right)\bar{u}_{i-1} + \left(1.0 + \frac{\Delta x_i}{\Delta x_i + \Delta x_{i-1}}\right)\bar{u}_i, \\ u_{i-1/2}^{(0)} &= \left(1.0 + \frac{\Delta x_i}{\Delta x_{i+1} + \Delta x_i}\right)\bar{u}_i - \left(\frac{\Delta x_i}{\Delta x_{i+1} + \Delta x_i}\right)\bar{u}_{i+1}, \\ u_{i-1/2}^{(1)} &= \left(\frac{\Delta x_i}{\Delta x_i + \Delta x_{i-1}}\right)\bar{u}_{i-1} + \left(\frac{\Delta x_{i-1}}{\Delta x_i + \Delta x_{i-1}}\right)\bar{u}_i. \end{aligned} \tag{7}$$

In the formulas (5), w_0, w_1 and $\tilde{w}_0, \tilde{w}_1$ are called nonlinear weights in the WENO methods, which are defined as

$$w_r = \frac{\alpha_r}{\alpha_1 + \alpha_2}, \quad \alpha_r = \frac{d_r}{(\epsilon + \beta_r)^2}, \quad r = 0, 1, \tag{8}$$

and

$$\tilde{w}_r = \frac{\tilde{\alpha}_r}{\tilde{\alpha}_1 + \tilde{\alpha}_2}, \quad \tilde{\alpha}_r = \frac{\tilde{d}_r}{(\epsilon + \beta_r)^2}, \quad r = 0, 1. \tag{9}$$

β_r is the smoothness indicator for the $(k-1)$-th (here, $k = 2$) degree reconstruction polynomial p_r associated with the small stencil $S_r = \{I_{i-r}, I_{i-r+1}\}$, while $d_r, \tilde{d}_r$ are called the optimal linear weights. The smoothness indicator is defined as

$$\beta_r = \sum_{l=1}^{k-1} \int_{x_{i-1/2}}^{x_{i+1/2}} \Delta x_i^{2l-1} \left(\frac{d^l p_r(x)}{dx^l}\right)^2 dx. \tag{10}$$

The explicit formulae on uniform grids for β_0 and β_1 are

$$\begin{aligned} \beta_0 &= (\bar{u}_{i+1} - \bar{u}_i)^2, \\ \beta_1 &= (\bar{u}_i - \bar{u}_{i-1})^2, \end{aligned} \tag{11}$$

and on nonuniform grids, they are

$$\begin{aligned}\beta_0 &= 4.0\left(\frac{\Delta x_i}{\Delta x_{i+1}+\Delta x_i}\right)^2(\bar{u}_{i+1}-\bar{u}_i)^2,\\ \beta_1 &= 4.0\left(\frac{\Delta x_i}{\Delta x_i+\Delta x_{i-1}}\right)^2(\bar{u}_i-\bar{u}_{i-1})^2.\end{aligned}\tag{12}$$

The optimal linear weights on uniform grids are

$$\begin{aligned}d_0 &= \frac{2}{3}, \quad d_1 = 1.0 - d_0 = \frac{1}{3},\\ \tilde{d}_0 &= \frac{1}{3}, \quad \tilde{d}_1 = 1.0 - \tilde{d}_0 = \frac{2}{3},\end{aligned}\tag{13}$$

and on nonuniform grids, they are

$$\begin{aligned}d_0 &= \frac{\Delta x_i + \Delta x_{i-1}}{\Delta x_{i+1}+\Delta x_i+\Delta x_{i-1}}, \quad d_1 = 1.0 - d_0 = \frac{\Delta x_{i+1}}{\Delta x_{i+1}+\Delta x_i+\Delta x_{i-1}},\\ \tilde{d}_0 &= \frac{\Delta x_{i-1}}{\Delta x_{i+1}+\Delta x_i+\Delta x_{i-1}}, \quad \tilde{d}_1 = 1.0 - \tilde{d}_0 = \frac{\Delta x_{i+1}+\Delta x_i}{\Delta x_{i+1}+\Delta x_i+\Delta x_{i-1}}.\end{aligned}\tag{14}$$

Lastly, ϵ is a small positive number chosen to avoid the denominator becoming 0. It is taken to be 10^{-6} in all our numerical tests.

2.2 Two-dimensional case

Consider the 2D hyperbolic equation for $u(x, y, t)$,

$$u_t + f(u)_x + g(u)_y = 0,\tag{15}$$

which is defined on the spatial domain $\Omega = [a, b] \times [c, d]$, with appropriate boundary condition and the initial condition $u(x, y, 0) = u_0(x, y)$. $f(u)$ and $g(u)$ are the flux functions. We partition the domain by the computational cells $I_{ij} = [x_{i-1/2}, x_{i+1/2}] \times [y_{j-1/2}, y_{j+1/2}]$ for $1 \le i \le N$ and $1 \le j \le M$, where $a = x_{1/2} < x_{3/2} < \cdots < x_{N+1/2} = b$, and $c = y_{1/2} < y_{3/2} < \cdots < y_{M+1/2} = d$. The center of the cell I_{ij} is (x_i, y_j), where $x_i = (x_{i-1/2} + x_{i+1/2})/2$ and $y_j = (y_{j-1/2} + y_{j+1/2})/2$. The grid sizes are $\Delta x_i = x_{i+1/2} - x_{i-1/2}$, $i = 1, \cdots, N$, and $\Delta y_j = y_{j+1/2} - y_{j-1/2}$, $j = 1, \cdots, M$. Using the cell averages of the function $u(x, y, t)$,

$$\bar{u}_{ij} = \frac{1}{\Delta x_i \Delta y_j}\int_{I_{ij}} u(x, y, t)dxdy, \quad i = 1, \cdots, N; j = 1, \cdots, M,\tag{16}$$

the finite volume scheme for the 2D problem (15) is formulated as:

$$\frac{d\bar{u}_{ij}(t)}{dt} = -\frac{1}{\Delta x_i}\left(\hat{f}_{i+\frac{1}{2},j} - \hat{f}_{i-\frac{1}{2},j}\right) - \frac{1}{\Delta y_j}\left(\hat{g}_{i,j+\frac{1}{2}} - \hat{g}_{i,j-\frac{1}{2}}\right).\tag{17}$$

The numerical flux $\hat{f}_{i+\frac{1}{2},j}$ is defined by

$$\hat{f}_{i+\frac{1}{2},j} = \sum_{\alpha} w_\alpha h\left(u^-_{i+\frac{1}{2},y_j+\beta_\alpha\Delta y_j}, u^+_{i+\frac{1}{2},y_j+\beta_\alpha\Delta y_j}\right),\tag{18}$$

for approximating the integration in y:

$$\frac{1}{\Delta y_j}\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} f(u(x_{i+\frac{1}{2}},y,t))dy,$$

where the function h is a monotone numerical flux and here we use the Lax-Friedrichs flux as in (3). β_α and w_α are Gaussian quadrature nodes and weights. The numerical flux $\hat{g}_{i,j+\frac{1}{2}}$ is defined similarly by

$$\hat{g}_{i,j+\frac{1}{2}} = \sum_\alpha w_\alpha h\left(u^-_{x_i+\beta_\alpha\Delta x_i,j+\frac{1}{2}}, u^+_{x_i+\beta_\alpha\Delta x_i,j+\frac{1}{2}}\right), \tag{19}$$

for approximating the integration in x:

$$\frac{1}{\Delta x_i}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} g(u(x,y_{j+\frac{1}{2}},t))dx.$$

Since for a n-point Gaussian quadrature, the approximation is exact for polynomials of degree $2n-1$ or less, a two-point Gaussian quadrature is sufficient for the WENO3 scheme here. We use the two-point Gaussian quadrature with the nodes $\beta_\alpha = \pm\frac{1}{2\sqrt{3}}$ and the weights $w_\alpha = \frac{1}{2}$, $\alpha = 0, 1$.

To compute the WENO reconstruction values at the Gaussian points $u^\pm_{i+\frac{1}{2},y_j+\beta_\alpha\Delta y_j}$ for $\hat{f}_{i+\frac{1}{2},j}$ and $u^\pm_{x_i+\beta_\alpha\Delta x_i,j+\frac{1}{2}}$ for $\hat{g}_{i,j+\frac{1}{2}}$, we adopt the dimension by dimension procedure as in [45]. This procedure involves two one-dimensional reconstructions described in the previous subsection, each one to remove a one-dimensional cell average in one of the two spatial directions. That is, we first perform a one-dimensional WENO reconstruction in the y-direction (or x-direction), in order to get the one-dimensional cell averages in the x-direction (or y-direction), i.e., the line averages of the numerical solutions at the cell boundaries,

$$u^\pm_{i,j+\frac{1}{2}} \approx \frac{1}{\Delta x_i}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u(x,y^\pm_{j+\frac{1}{2}},t)dx \text{ (or } u^\pm_{i+\frac{1}{2},j} \approx \frac{1}{\Delta y_j}\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} u(x^\pm_{i+\frac{1}{2}},y,t)dy).$$

Then we perform another one-dimensional WENO reconstruction using these one-dimensional cell averages in the x-direction (or y-direction), to obtain the final reconstructed point values at the Gaussian points. Note that the optimal linear weights in 2D finite volume WENO schemes may have negative values, as discussed in [45]. In such case, the splitting technique in [45] provides one way to resolve it.

Remark 1 For the cases of systems, the third-order finite volume WENO method is applied in a component by component fashion. We make the reconstruction using the scalar WENO3 procedure for each of the components of u separately to obtain the left and right values at a cell interface. The Lax-Friedrichs flux is then used to form the scheme (4) or (17). This component-wise version of WENO scheme for solving a system of equations is simple and cost effective. It works well when the order of accuracy is not high, such as the third-order scheme considered in this paper. However, when a

WENO scheme with higher order of accuracy is used, which will be studied in our future work, the more costly, but more robust characteristic-wise implementation will be adopted [47].

3 A multilayer perceptron troubled-cell indicator

In this section, we describe the procedure to construct the MLP model for the troubled-cell indicator.

3.1 The MLP model

An artificial neural network method is a class of machine learning methods that consists of nonlinear functions with parameterizable coefficients. Multiple simple processing units, which are called neurons, are combined together into a large parallel distributed neural network system [14]. Each of these neurons is available to store knowledge during learning processes. Given an input vector of data $\mathbf{v}$, a neural network maps it to an output $\hat{\mathbf{y}} = f(\mathbf{v}, \Theta)$, where Θ is a collection of learnable parameters. The links to connect input and output of neural networks are multiple hidden layers of neurons. The vector of output of a previous hidden layer $\mathbf{u}^{k-1}$ is taken as the input of the current hidden layer. These connecting links to obtain the output vector $\mathbf{u}^k$ of the current hidden layer, denoted by a mapping $\mathbf{L}^k$, are defined as

$$\mathbf{u}^k = \mathbf{L}^k(\mathbf{u}^{k-1}) = \phi(\mathbf{w}_k^T \mathbf{u}^{k-1} + \mathbf{b}_k). \tag{20}$$

The mapping $\mathbf{L}^k$ is characterized by the parameters including the weights $\mathbf{w}_k \in \Theta$ and the bias $\mathbf{b}_k \in \Theta$ which are stored in a matrix and a vector respectively, and an activation function $\phi(z)$ which is a scalar function. The activation function ϕ is applied element-wise to each element of the computed vector $\mathbf{w}_k^T \mathbf{u}^{k-1} + \mathbf{b}_k$, to obtain the vector $\mathbf{u}^k$. Combining multiple hidden layers, for example K layers, we have the network output

$$\hat{\mathbf{y}} = \mathbf{L}^K(\cdots(\mathbf{L}^2(\mathbf{L}^1(\mathbf{v})))). \tag{21}$$

Among various neural network models, we focus on the multilayer perceptron (MLP) model, which was introduced by Rosenblatt in 1958 [43]. It is one of the simplest and the most common deep neural network models, and consists of an input layer, one or more hidden layers, and an output layer. Adjacent layers in a MLP model are fully connected, so that every node in each layer of the network is connected to every other node in the adjacent layers. Properties of the MLP's universal approximation to nonlinear functions were discussed in e.g. [10, 16]. A properly trained MLP model is able to detect different types of smooth functions and discontinuities, and has a straightforward architecture which contributes to its low cost of training and computational procedure compared to other more complex neural network models, for example, convolution neural networks (CNNs) [26]. Motivated by [49] where the CNN is used to develop a troubled-cell indicator for finite difference WENO schemes, and the approaches in [41, 61, 4, 59] where the MLP is used to help distinguish nonsmooth and smooth regions in discontinuous Galerkin methods or finite difference WENO schemes,

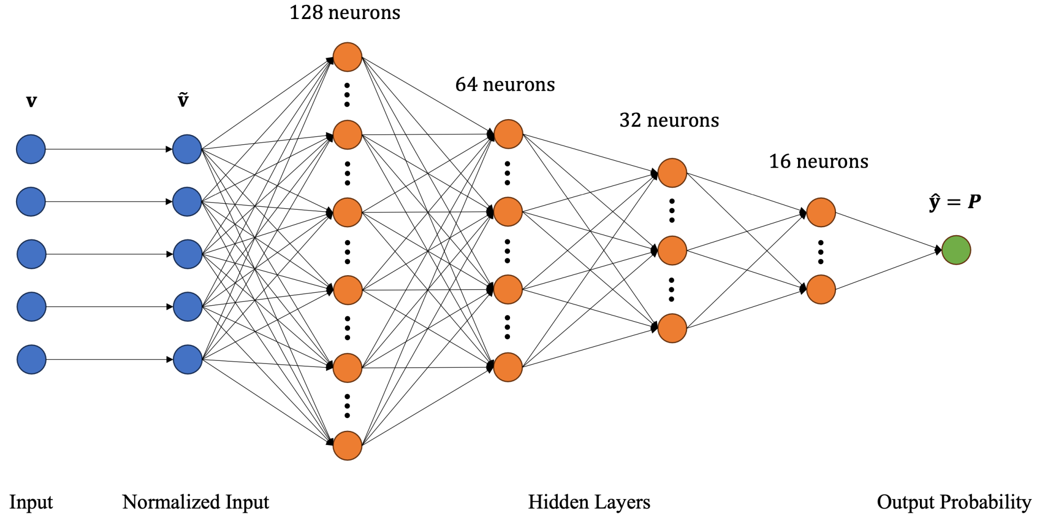

Figure 1: Diagram of the constructed MLP model architecture with an input layer, a normalization layer, four hidden layers, and an output layer.

here we construct a MLP based troubled-cell indicator and obtain a third-order hybrid finite volume WENO scheme. Fig. 1 displays the diagram of the MLP model we construct, which consists of an input layer to take a five-dimensional input data, a normalization layer to standardize the input, four fully connected hidden layers, and an output layer to predict a probability value. In the following, the detailed procedure is described.

3.2 Dataset

As the design of training and validation data is crucial in training neural network models, we first describe how the dataset for the network training of this MLP model is generated. For a target cell I_i, the input dataset contains a combination of cell average values and point values of functions in this cell and its neighbors. Specifically, similar to [41], the generated input dataset for the MLP model here is $\mathbf{v}_i = (\bar{u}_{i-1}, \bar{u}_i, \bar{u}_{i+1}, u^-_{i+1/2}, u^+_{i-1/2}) \in \mathbb{R}^5, \forall i$. For the neural network training, we follow a supervised learning procedure [27]. That is, the exact output results are included in the dataset so that the model has a baseline knowledge of the correct output results. During the network training, the model maps the input dataset to the predicted output results, which are compared with these exact outputs. The loss function of the model is minimized in the supervised learning such that accurate predictions are made for the general data. In the MLP model training here for a troubled-cell indicator, the output represents the probability about the smoothness of the solution function on a target cell. Hence we generate the exact output dataset where the output is labeled either 0 if a function on the target cell I_i has a singularity or 1 if the function on the target cell I_i is smooth.

The input dataset is composed of two types of data: function values from canonical functions and numerical solutions from the finite volume WENO3 scheme, which are summarized in Table 1. As in [61, 4], the type I training data are function values from piecewise functions such as step functions and trigonometric functions, which are smooth or contain discontinuities. The neural network learns different scenarios of smooth and nonsmooth functions. For a function used to generate the type I training data, we randomly pick a point in the domain as the center of the target cell I_i and a mesh size h using uniform distributions, such that the target cell I_i and its two neighboring cells I_{i-1} and I_{i+1} are inside the domain. Three cell average values $\bar{u}_{i-1}$, $\bar{u}_i$, $\bar{u}_{i+1}$ and two cell boundary values $u^-_{i+1/2}$, $u^+_{i-1/2}$ are computed, which serve as the input training data for the target cell I_i in this domain and one of the samples in the input dataset. Note that for the absolute value functions and the step functions, we only include the samples of nonsmooth cases (i.e., the function has a singularity on one of the cells I_{i-1}, I_i, I_{i+1}) in the dataset. Via this way, enough data of nonsmooth cases are provided to the neural network to learn such that the model is more accurate in identifying troubled-cells.

The type II training data are numerical solutions of the WENO3 scheme solving the linear advection equation $u_t + au_x = 0$ with various initial conditions of piecewise functions, as in [49, 61]. The impact of the numerical method on the structure of the solution, for example controlling numerical oscillations near a discontinuity or the smearing caused by the numerical viscosity, is taken into consideration. The model learns the feature of the numerical solutions of the finite volume WENO3 scheme from the type II data. The procedure to generate the data is presented here. For an advection coefficient a randomly selected from the uniform distribution $U[-1, 1]$, we first generate the piecewise smooth initial condition $u_0(x)$ of the linear advection equation $u_t + au_x = 0$ as follows:

1. Randomly select 4 locations for the discontinuities of $u_0(x)$ in the domain $[-1, 1]$ using the uniform distribution, to divide the domain into 5 subdomains.
2. Within the most left and the most right subdomains, the initial condition $u_0(x)$ is set to be 0. Within each of the other three subdomains, the initial condition $u_0(x)$ is created using random Fourier series $a_0 + \sum_{n=1}^{N_f}(a_n \cos(nx) + b_n \sin(nx))$, where N_f is randomly picked from the discrete uniform distribution $U\{1, 6\}$. The Fourier coefficients a_0, a_n and b_n are randomly selected from the standard normal distribution $N(0, 1)$.

Using the periodic boundary condition, we apply the third-order finite volume WENO scheme in Section 2.1 on a grid with the number of computational cells $N = 320$, to solve the equation. The numerical solution is calculated for N_t time steps, where N_t is chosen from the discrete uniform distribution $U\{1, 10\}$. At the final time T, we collect the data from the numerical solution for each cell I_i: $\mathbf{v}_i = (\bar{u}_{i-1}, \bar{u}_i, \bar{u}_{i+1}, u^-_{i+1/2}, u^+_{i-1/2})$, which provides one of the samples for the type II data of the neural network's input dataset. In the type II data, the exact output data is collected based on the exact solution $u(x, T) = u_0(x - aT)$. Namely, if the exact solution has a discontinuity in the target cell I_i, the output data is labeled 0, otherwise if the exact solution is smooth on I_i, the output is labeled 1.

Training dataset			
$u(x)$	Domain	Varied parameters	# of samples
$a\|x\|$	$[-0.5, 0.5]$	$a \in U[1, 10]$	3200
$u_l \mathbf{1}_{x<a} + u_r \mathbf{1}_{x>a}$	$[-1, 1]$	$(u_l, u_r) \in U[-4, 4]$, $a \in U[-0.56, 0.56]$	10240
$\sin(k\pi x)$	$[0, \frac{k}{4}]$	$k \in U[1, 25]$	20480
$(\sum_{n=0}^{2} b_{n,l} \sin(n\pi x) + c_{n,l} \cos(n\pi x))\mathbf{1}_{x<a}$ $+(\sum_{n=0}^{2} b_{n,r} \sin(n\pi x) + c_{n,r} \cos(n\pi x))\mathbf{1}_{x>a}$	$[-1, 1]$	$(b_{n,\cdot}, c_{n,\cdot}) \in U[-5, 5]$ $a \in U[-0.56, 0.56]$	4480
$\tanh(kx)$	$[-1, 1]$	$k \in U[5, 30]$	4480
$k \sin(2\pi x) \cos(3\pi x) \sin(4\pi x)$	$[0, 1]$	$k \in U[0.2, 2]$	4480
$k \sin^4(\pi x)$	$[0, 1]$	$k \in U[0.5, 3]$	4480
Numerical solutions of $u_t + au_x = 0$	$[-1, 1]$	$a \in U[-1, 1]$	16640

Table 1: Functions and numerical solutions used to generate training data. U represents a uniform distribution. $\mathbf{1}_{(\cdot)}$ represents an indicator function.

3.3 Network architecture

In this section, we describe the architecture of the MLP neural network model shown in Fig. 1. The MLP model takes a 5-dimensional input vector $\mathbf{v}$. Following the procedure in [61], let v_l denote the l-th element in the input vector. We normalize the generated data v_l, $l = 1 \dots 5$ before feeding it into the hidden layers of the model as the following

$$\tilde{v}_l = \frac{v_l - \mu}{\sigma}, \tag{22}$$

where μ and σ are respectively the mean and the standard deviation of the elements of all data $\mathbf{v}$ in the dataset. The dataset is randomly split into a training set and a validation set with 8 : 2 proportion.

In the MLP model, aside from an input layer, a normalized input layer, and an output layer, there are four hidden layers that contain 128, 64, 32, and 16 neurons sequentially. Since there are no existing universal rules on choosing number of neurons for deep learning models, here we follow the practice in [61] in setting up these numbers of neurons. Note that fewer number of neurons in the subsequent layers than the previous layers could be more cost-effective in training classification problems, as shown in [56]. As suggested in [13], the weights of the model are randomly initialized using the Xavier normal distribution $\mathcal{N}(0, \mathrm{std}_w^2)$ with $\mathrm{std}_w = \sqrt{\frac{2}{n_j + n_{j+1}}}$, and the biases are randomly initialized using the uniform distribution $U[-\mathrm{std}_b, \mathrm{std}_b]$ with $\mathrm{std}_b = \sqrt{\frac{1}{n_j}}$. Here n_j is the number of incoming neurons of the current layer and n_{j+1} is the number of outgoing neurons of the current layer. For example, n_j and n_{j+1} are 5 and 128 respectively for the first hidden layer, and 128 and 64 respectively for the second hidden layer, and so forth in the model. Within each hidden layer except the first hidden layer, Leaky Rectified Linear Unit (Leaky ReLU)

$$\phi(z) = \begin{cases} z, & \text{if } z \geq 0, \\ 0.01z, & \text{otherwise} \end{cases} \tag{23}$$

is used as the activation function [36]. For the first hidden layer, hyperbolic tangent function Tanh is used as the activation function. The output layer has a single neuron. For the output layer, Sigmoid

$$\phi(z) = \frac{1}{1 + e^{-z}} \tag{24}$$

is used as the activation function since the output is a value of probability ranging from 0 to 1 [44]. The output value of the neural network represents the predicted probability that the function on a target cell is smooth. The difference between this predicted probability value $\hat{\mathbf{y}}$ and the referenced label $\mathbf{y}$ (the exact indication about the smoothness of the function on the target cell) in the output dataset is measured by a loss function. Note that the goal of training a model is to minimize the loss function and an appropriate loss function should be chosen. Here the output is a binary classification on whether the function on a cell is smooth or not, so the Binary Cross Entropy (BCELoss)

$$-\frac{1}{S_b} \sum_{i=1}^{S_b} (y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)) \tag{25}$$

is used as the loss function [9]. $\hat{y}_i$ is the model output for the input sample $\mathbf{v}_i$, y_i is the reference label for the input sample $\mathbf{v}_i$, and S_b is the mini-batch size, which is the number of samples utilized in one iteration of the training or validation procedure. Note that the training or validation dataset is divided into several mini batches. If the number of samples of the entire dataset is not an integer multiple of mini-batch size, the last mini-batch will contain the remaining samples as a smaller batch. The mini-batch size $S_b = 500$ is used for this MLP model here. To update the weights and bias parameters of the MLP model by minimizing the loss function, we use the Adam optimizer [22] with the learning rate $\eta = 0.0001$ and the weight decay $r_d = 0.00008$. It is a variation of stochastic gradient descent and has been commonly used in training neural network models since its introduction, due to its fast convergence and good performance for a trained model not to overfit and to make accurate predictions on new data that are not in the training dataset. We choose the epoch number epo $= 100$, so every sample of the dataset is fed to the model for 100 times and the learnable parameters of the model are updated. The final MLP model obtained after the whole training and validation procedure has 98.79% accuracy on the training data and 96.95% accuracy on the validation data. This indicates that the obtained MLP model is not overfitted on the training dataset and gives good predictions on the whole dataset.

4 The hybrid finite volume WENO3 scheme with the MLP troubled-cell indicator

In this section, we describe the hybrid finite volume WENO3 scheme with the MLP troubled-cell indicator, which is called the WENO3-MLP scheme in this paper. This hybrid scheme is based on the finite volume WENO3 scheme in Section 2 and the MLP model obtained in Section 3. The WENO3-MLP scheme for a 1D problem is described as follows:

1. Define two hyper-parameters: the probability threshold P and the parameter N_b related to the size of the buffer zone (see Remark 2 for discussions on the buffer zone). In our simulations, the default choice is $P = 0.9, N_b = 3$.
2. Use the PDE's initial condition $u_0(x)$ to generate the initial input data of the neural network
$$\mathbf{v}_i = (\bar{u}_{i-1}^{(0)}, \bar{u}_i^{(0)}, \bar{u}_{i+1}^{(0)}, u_{i+1/2}^{-,(0)}, u_{i-1/2}^{+,(0)})$$
for all i. Here the initial values on cell boundary points $u_{i+1/2}^{-,(0)}, u_{i-1/2}^{+,(0)}$ are directly evaluated using the initial condition $u_0(x)$.
3. In the current time step and the current stage of the third-order TVD Runge-Kutta (TVD-RK3) [48] time evolution, for every target cell $I_i,\ i = 1, 2, \cdots$: feed the neural network input data into the MLP model to obtain the output P_i, where P_i represents the probability that the numerical solution on the cell I_i is considered as smooth, and a larger value of P_i informs that the numerical solution on the cell I_i is more likely to be smooth.
4. For $i = 1, 2, \cdots$:
   (a) If any of the probability outputs for the target cell I_i and its neighboring cells in the buffer zone $\{I_{i-N_b+k},\ k = 0, 1, \cdots, 2N_b\}$ is less than the threshold value P (i.e. if $P_l < P$ for any $l \in \{i - N_b + k,\ k = 0, 1, \cdots, 2N_b\}$), the WENO3 reconstruction (5) is applied for computing the numerical values $u_{i+1/2}^-$ and $u_{i-1/2}^+$ on the cell boundaries of I_i.
   (b) Else, the third-order linear reconstruction (i.e., replace the nonlinear weights in (5) by the linear weights) is applied for computing the numerical values $u_{i+1/2}^-$ and $u_{i-1/2}^+$ on the cell boundaries of I_i.

   Denote the obtained reconstruction values $u_{i+1/2}^-,\ u_{i-1/2}^+$ by $u_{i+1/2}^{-,*},\ u_{i-1/2}^{+,*}$ for the convenience of description in the following step.
5. For $i = 1, 2, \cdots$: compute the Lax-Friedrichs numerical flux (3) and the spatial discretization in the finite volume scheme (4), and use the TVD-RK3 scheme to obtain the new cell average value $\bar{u}_i^{(new)}$ at the next stage of TVD-RK3. For all i, form the neural network input data $(\bar{u}_{i-1}^{(new)}, \bar{u}_i^{(new)}, \bar{u}_{i+1}^{(new)}, u_{i+1/2}^{-,*}, u_{i-1/2}^{+,*})$ and use it to do the WENO3-MLP reconstruction (the steps 3 and 4) again to update the cell boundary values $u_{i+1/2}^{-,*},\ u_{i-1/2}^{+,*}$, which provide accurate cell boundary values in the neural network input data for the next stage of TVD-RK3 and are named as $u_{i+1/2}^{-,(new)},\ u_{i-1/2}^{+,(new)}$. It is found in the numerical experiments that

this additional WENO3-MLP reconstruction is necessary for obtaining smaller numerical errors and uniform accuracy orders of the proposed hybrid scheme, which is emphasized further in the following Remark 3.

6. Use the neural network input data $(\bar{u}_{i-1}^{(new)}, \bar{u}_i^{(new)}, \bar{u}_{i+1}^{(new)}, u_{i+1/2}^{-,(new)}, u_{i-1/2}^{+,(new)})$ obtained in the step 5 and repeat the steps 3 to 5 for the next stage of the TVD-RK3 scheme, and finish the computation of the current time step.
7. Repeat the steps 3 to 6 until the final time T is reached.

Remark 2 The buffer zone is used in the WENO3-MLP scheme for applying the WENO reconstructions on the troubled-cells identified by the neural network and their neighboring cells. It is a small neighborhood of a target cell which includes the target cell and the neighboring cells of the target cell. The size of a buffer zone is preset. If any cell in the buffer zone of a target cell is indicated as a troubled-cell, then WENO3 reconstruction is applied on that target cell. This approach of using buffer zones has been proposed to achieve better nonlinear stability in developing various hybrid schemes for hyperbolic conservation laws, including both the traditional methods (e.g. [7, 8, 55]) and the methods using the neural network models (e.g. [49, 56]).

Remark 3 We emphasize the importance of the additional WENO3-MLP reconstruction in Step 5 of the WENO3-MLP scheme described above, for accurate indication of troubled-cells by the MLP model. In the numerical experiments (Example 6.1.1 and Example 6.1.2) of Section 6, we compare the numerical results of the scheme with and without doing the additional WENO3-MLP reconstruction in Step 5. It is verified that more accurate numerical solutions and uniform convergence orders are obtained if such additional WENO3-MLP reconstruction is used in the scheme. Also, it is shown in the numerical results that this additional reconstruction has very little impact on the computational cost of the whole simulation.

The same MLP model, which is trained on 1D data, is directly applied in the dimension by dimension reconstruction procedure for two-dimensional problems. This is similar to the approaches in [61, 56]. It is different from the neural network based troubled-cell indicators for which the models to solve two-dimensional problems are separately trained on 2D data [41, 42, 53]. With the default hyper-parameters $P = 0.9$ and $N_b = 3$ and the initial condition $u_0(x, y)$, we follow the similar procedure of Step 2 in the WENO3-MLP scheme for 1D problems to generate the neural network input data $\mathbf{v}_{ij}^x$ and $\mathbf{v}_{ij}^y$ of the x-direction and the y-direction respectively for the MLP model to solve a 2D problem, which are

$$\mathbf{v}_{ij}^x = (\bar{u}_{i-1,j}, \bar{u}_{ij}, \bar{u}_{i+1,j}, u_{i+1/2,j}^-, u_{i-1/2,j}^+),$$
$$\mathbf{v}_{ij}^y = (\bar{u}_{i,j-1}, \bar{u}_{ij}, \bar{u}_{i,j+1}, u_{i,j+1/2}^-, u_{i,j-1/2}^+)$$

for all i, j. Here $u_{i+1/2,j}^-, u_{i-1/2,j}^+$ and $u_{i,j+1/2}^-, u_{i,j-1/2}^+$ are line averages of the numerical solutions at the cell boundaries. Perform Step 3 and Step 4 of the WENO3-MLP algorithm for 1D problems, in both the x-direction and the y-direction to reconstruct line average values, and these values at the Gaussian quadrature points in the numerical fluxes $\hat{f}$ and $\hat{g}$ in (18) and (19). Then, similar to Step 5 in the 1D algorithm, we form the 2D finite volume scheme (17) and march one stage of the TVD-RK3 scheme to obtain the new cell average values at the next stage. An additional WENO3-MLP reconstruction is performed to reconstruct and update line average values at cell boundaries $u^{\pm}_{i+1/2,j}$, $u^{\pm}_{i,j+1/2}$ for all i, j, to provide accurate line average values at the cell boundaries in the input data of the MLP model for the next stage of TVD-RK3 scheme, as that in Step 5 of the 1D algorithm. Repeat the procedure for the next stages to finish the computation of the current time step, and march forward until the final time T is reached.

Similarly, for the cases of systems, this third-order hybrid finite volume WENO method is applied in a component by component fashion.

5 The KXRCF troubled-cell indicator

In this section, we briefly review one of the most popular traditional troubled-cell indicators in the literature, the KXRCF indicator proposed in [23]. The KXRCF indicator has been shown in [40, 31] to be one of the best performed troubled-cell indicators. To study the proposed MLP troubled-cell indicator in this paper, we carry out extensive numerical tests to compare the KXRCF indicator with the MLP troubled-cell indicator in the next section. In general, for the target cell I_i, the KXRCF indicator value κ_i of a third-order scheme is defined as

$$\kappa_i = \frac{\left|\int_{\partial I_i^-} (u_h|_{I_i} - u_h|_{I_{nb,i}}) ds\right|}{h^{\frac{3}{2}} |\partial I_i^-| \|u_h\|_{I_i}}, \tag{26}$$

where h is the radius of the circumscribed circle in the cell I_i, ∂I_i^- is the inflow portion of the cell boundary, $I_{nb,i}$ is the neighbor of I_i on the side of ∂I_i^-, and $|\partial I_i^-|$ is the length (area) of ∂I_i^-. u_h is the numerical solution. For a 1D problem, the norm $\|u_h\|_{I_i}$ in the denominator is based on the cell average value on I_i [23, 40]. For a 2D problem, the KXRCF indicator value κ_{ij} of the target cell I_{ij} is formulated as

$$\kappa_{ij} = \frac{\left|\int_{\partial I_{ij}^-} (u_h|_{I_{ij}} - u_h|_{I_{nb,ij}}) ds\right|}{\left((\frac{\Delta x_i}{2})^2 + (\frac{\Delta y_j}{2})^2\right)^{\frac{3}{4}} |\partial I_{ij}^-| \|u_h\|_{I_{ij}}}, \tag{27}$$

where ∂I_{ij}^- is the inflow portion of the cell boundary of I_{ij}, $I_{nb,ij}$ are neighboring cells sharing the edges ∂I_{ij}^-, and $|\partial I_{ij}^-|$ is the length of ∂I_{ij}^-. u_h is the numerical solution, and the norm in the denominator here is the maximum norm taken at the integration quadrature points [23, 40]. The integral in the numerator of (27) is computed by first evaluating the absolute value of the difference of the line averages on two sides for each of the edges in ∂I_{ij}^-, and multiplying it with the length of the corresponding edge, then summing up these values over all edges of ∂I_{ij}^-. The target cell I_i (or I_{ij}) is marked "smooth" if $\kappa_i \leq 1$ (or $\kappa_{ij} \leq 1$), or "troubled" if $\kappa_i > 1$ (or $\kappa_{ij} > 1$).

We also apply the buffer zone for implementing the KXRCF indicator as that in [49]. Similar to [49], it is observed that slight oscillations appear near discontinuities of a solution in some numerical examples, if the buffer zone is not used. Here for the consistency, we adopt the same parameter of the buffer zone $N_{b,\kappa} = 3$ for the KXRCF indicator as that for the MLP troubled-cell indicator. The KXRCF troubled-cell indicator is utilized to form a hybrid WENO3 scheme via simply adding a calculation of the indicator values and an *if* statement before computing the reconstruction values on target cells' boundaries. Namely, for the target cell I_i in a 1D problem, if $\kappa_i > 1$ or $\kappa_{i-1} > 1$ or $\kappa_{i+1} > 1$ or $\kappa_{i-2} > 1$ or $\kappa_{i+2} > 1$ or $\kappa_{i-3} > 1$ or $\kappa_{i+3} > 1$, the WENO3 scheme is applied for the reconstructions on this target cell; otherwise, the third-order linear scheme is used for the reconstructions. For a 2D problem, the dimension by dimension procedure is adopted. Hence, for the target cell I_{ij}, the same *if* statement with the conditions for the cells in the buffer zone to determine troubled-cells as that in a 1D problem is used in the reconstructions for each spatial direction. For example in the x-direction, if $\kappa_{ij} > 1$ or $\kappa_{i-1,j} > 1$ or $\kappa_{i+1,j} > 1$ or $\kappa_{i-2,j} > 1$ or $\kappa_{i+2,j} > 1$ or $\kappa_{i-3,j} > 1$ or $\kappa_{i+3,j} > 1$, the WENO3 scheme is applied for the reconstructions of the x-direction for this target cell; otherwise, the third-order linear scheme is used.

6 Numerical experiments

In this section, we solve a series of 1D and 2D numerical examples on both uniform grids and nonuniform grids using the proposed WENO3-MLP scheme, and compare it with the classical finite volume WENO3 scheme described in Section 2 and the hybrid WENO3 scheme using the KXRCF troubled-cell indicator in Section 5. The accuracy orders for solving problems with smooth solutions, the ability to handle oscillations for problems with nonsmooth solutions, and computational cost of these different schemes are compared. In the comparison of the MLP troubled-cell indicator and the KXRCF troubled-cell indicator, we show the cells being flagged as troubled-cells by each of these indicators. The CFL number is taken to be 0.5 in all tests. One of the advantage properties of the finite volume method is its flexibility on different grids rather than only uniform grids. To show that, we solve these examples on both uniform and nonuniform grids. For the 1D problems of testing accuracy orders of the schemes, the setup of the nonuniform grids and their refinements is as the following:

1. Denoting N as the number of computational cells, we start with $N = 20$ for the base grid whose nonuniform grid points are perturbed from the corresponding uniform grid points x_i by $\hat{x}_i = x_i + \delta_i$. δ_i is a random number in $[-ch, ch]$, where c is a constant chosen to adjust the perturbation strength and h is the uniform grid size with $N = 20$.
2. A coarse nonuniform mesh is refined in a uniform way such that the numerical accuracy orders of the schemes observed in the mesh refinement study approximate the schemes' theoretical accuracy order well. In the implementation, we take the center of two neighboring grid points of a coarse mesh as the newly added grid point of the refined mesh in the next level, while all grid points of the coarse mesh are still kept as the grid points of the refined mesh. So, the grid sizes of a refined mesh are exactly one half of the coarser one of its last level.

In general, for a problem without the need of mesh refinement, we randomly perturb the grid points of a uniform grid in the range $[-ch, ch]$ directly to obtain the nonuniform grid. The setup for a 2D nonuniform grid is similar to the 1D cases. For all i, j, the grid points x_i in the x-direction and y_j in the y-direction of a uniform grid are randomly perturbed independently by random numbers $\delta_{x,i} \in [-ch_x, ch_x]$ and $\delta_{y,j} \in [-ch_y, ch_y]$ respectively. c is the constant chosen to adjust the perturbation strength, and h_x, h_y are the grid sizes of the uniform grid in the x-direction and the y-direction respectively. In this paper, numerical experiments are conducted using different values of c for all examples. Since the numerical results using different values of c are consistent and similar conclusions are drawn, only the results using the nonuniform grids with $c = 0.4$ are shown in this section, to save spaces. As an illustration, Fig. 2 shows examples of the 1D and 2D nonuniform grids with $c = 0.4$ for the domain $[0, 2\pi]$ with $N = 20$, and for the domain $[0, 2\pi]^2$ with $N \times N = 20 \times 20$ respectively.

In this paper, we use the Python programming language and the same coding style to implement all schemes for solving all numerical examples, so that fair comparisons are performed for the proposed hybrid finite volume WENO3-MLP scheme and the other two numerical schemes. The Python programming language is chosen here to be consistent with the implementation of neural network model using the PyTorch package [38]. We are aware that other programming languages may also be used for the implementation, e.g. Fortran, C++, etc., and different compiler optimization flagging strategies and certain coding styles in which all arrays are vectorized for instance can be utilized to improve the efficiency of simulations. This is one of the interesting topics to be explored further in our next research.

6.1 Numerical examples with smooth solutions

Example 6.1.1 1D and 2D linear advection equations

(a) 1D case. We solve the 1D linear advection equation

$$u_t + u_x = 0 \tag{28}$$

on both uniform grids and nonuniform grids, with the initial condition $u_0(x) = \sin(x)$ and periodic boundary conditions. The computational domain is $[0, 2\pi]$ and the final time $T = 1.0$. Table 2 shows the accuracy table on uniform grids. It is observed that even on quite coarse meshes, the third-order accuracy has been achieved for both L_1 and L_∞ errors of the WENO3-MLP scheme and the hyrid WENO3 with the KXRCF indicator, whereas very refined mesh such as $N = 320$ has to be used to observe the third-order accuracy for the classical WENO3 scheme. We record and compare the computational costs (in seconds) of different schemes. It is observed that when the meshes are relatively coarse ($N = 20, 40, 80$), the WENO3-MLP takes more CPU time to compute the solution than the classical WENO3, due to the fact that the initialization of the neural network model takes majority of the whole simulation time in the WENO3-MLP. It is also due to the fact that the WENO3-MLP mis-identifies some of the cells as troubled-cells and the WENO3 procedure is used for these cells on which the solution is smooth. However when the meshes get more refined, the WENO3-MLP identifies all cells correctly as smooth cells (i.e., the cells which are not

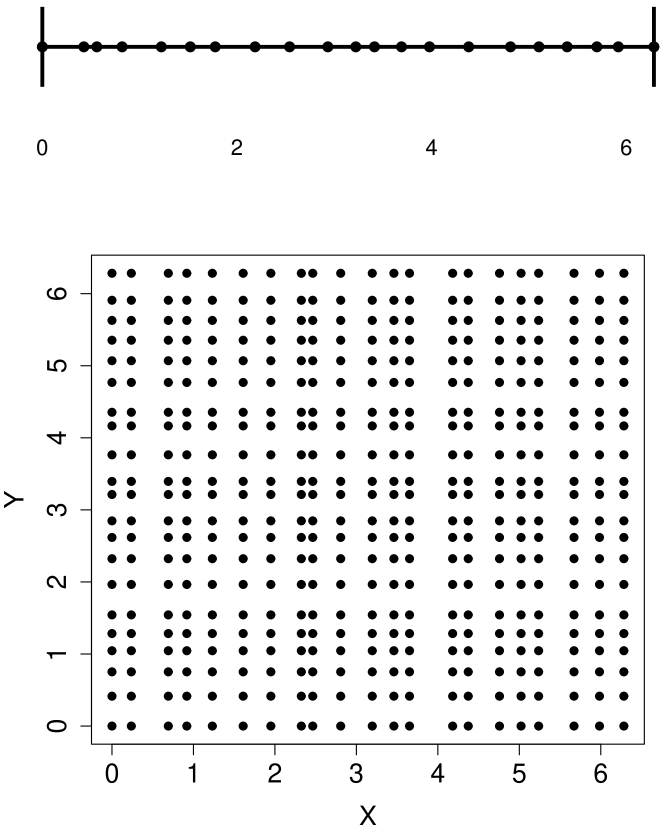

Figure 2: Examples of 1D and 2D nonuniform grids with $c = 0.4$. Top: a 1D nonuniform grid on $[0, 2\pi]$ with the number of cells $N = 20$; bottom: a 2D nonuniform grid on $[0, 2\pi]^2$ with the number of cells $N \times N = 20 \times 20$.

WENO3-MLP					
N	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU time (s)
20	2.30798e-02		6.90951e-02		0.76
40	9.64676e-04	4.580	3.21088e-03	4.428	0.83
80	3.87302e-05	4.639	8.75042e-05	5.197	0.93
160	4.69101e-06	3.045	7.36734e-06	3.570	1.19
320	5.86146e-07	3.001	9.20718e-07	3.000	2.39
640	7.32561e-08	3.000	1.15070e-07	3.000	6.77

Classical WENO3					
N	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU time (s)
20	2.98265e-02		6.90947e-02		0.04
40	7.63167e-03	1.997	2.65716e-02	1.379	0.13
80	1.85271e-03	2.042	1.07448e-02	1.306	0.49
160	3.75398e-04	2.303	3.49914e-03	1.619	1.85
320	4.19162e-05	3.163	6.25910e-04	2.483	7.23
640	2.59471e-06	4.014	3.83734e-05	4.028	28.86

WENO3 with KXRCF					
N	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU time (s)
20	2.38059e-03		3.74229e-03		0.03
40	2.99600e-04	2.990	4.71162e-04	2.990	0.08
80	3.75368e-05	2.997	5.89043e-05	3.000	0.26
160	4.69101e-06	3.000	7.36734e-06	2.999	0.94
320	5.86146e-07	3.001	9.20718e-07	3.000	3.55
640	7.32561e-08	3.000	1.15070e-07	3.000	14.14

Table 2: Example 6.1.1, 1D linear advection equation with the initial condition $u_0 = \sin(x)$ on uniform grids, and the smooth solution at $T = 1.0$. L_1 errors, L_∞ errors, numerical accuracy orders, and CPU time in seconds.

troubled-cells) so that only the linear scheme is used, which leads to much less CPU time than the WENO3 reconstruction procedure. It is observed that the WENO3-MLP is more than three times faster and more than four times faster than the classical WENO3 at $N = 320$ and $N = 640$ respectively. In the comparison with the WENO3 with the KXRCF indicator, we see that the numerical errors of the WENO3 with KXRCF indicator are smaller than those of the classical WENO3 for all Ns, and the same as those of the WENO3-MLP when the meshes are relatively refined. It is also noted that the WENO3-MLP requires much less computational time on refined meshes compared to the WENO3 with the KXRCF indicator, which requires additional costs to evaluate the indicator value κ_i. For example, on the mesh with $N = 640$, the WENO3-MLP is more than two times faster than the WENO3 with the KXRCF indicator. To highlight Remark 3 in Section 4 where the importance of the additional WENO3-MLP reconstruction in Step 5 of the WENO3-MLP scheme is emphasized, here we show the

N	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU time (s)
20	2.75434e-02		6.90952e-02		0.74
40	3.20905e-03	3.101	2.17636e-02	1.667	0.81
80	5.86812e-04	2.451	7.25628e-03	1.585	0.97
160	9.97840e-05	2.556	2.06976e-03	1.810	1.46
320	9.66993e-06	3.367	3.11337e-04	2.733	3.29
640	6.08078e-07	3.991	1.53343e-05	4.344	9.91

Table 3: Example 6.1.1, 1D linear advection equation with the initial condition $u_0 = \sin(x)$ on uniform grids, and the smooth solution at $T = 1.0$. WENO3-MLP scheme without the additional WENO3-MLP reconstruction for updated cell boundary values $u^-_{i+1/2}, u^+_{i-1/2}$ in the neural network input data. L_1 errors, L_∞ errors, numerical accuracy orders, and CPU time in seconds.

numerical results at $T = 1.0$ on uniform grids using the same WENO3-MLP scheme without the additional WENO3-MLP reconstruction for updated cell boundary values $u^-_{i+1/2}, u^+_{i-1/2}$ in the neural network input data. The numerical results are shown in Table 3. It is verified that without such additional WENO3-MLP reconstruction, numerical errors on all grids are larger than those of the WENO3-MLP scheme reported in Table 2, and the third-order accuracy is observed only when the mesh is quite refined. The CPU times on most of the grids are larger than those reported in Table 2 for the WENO3-MLP, because more cells are mis-identified as troubled-cells on which the more costly WENO3 reconstruction scheme is used instead of the linear scheme. It is shown in this example that the additional WENO3-MLP reconstruction for obtaining updated cell boundary values in the neural network input data improves the accuracy of the MLP model for indication of the troubled-cells, and actually makes the whole WENO3-MLP scheme more efficient. Similar results are obtained on nonuniform grids with $c = 0.4$. Table 4 shows the results of the accuracy, convergence orders, and computational costs of these three schemes. The similar advantages of the WENO3-MLP scheme to those on uniform grids are observed. Note the computational times on nonuniform grids are longer than the computational times on uniform grids for the same N and the same numerical scheme. It is caused by the smaller time step sizes, which are determined by the CFL condition and the fact that the smallest mesh size $\min_i \Delta x_i$ is smaller on the nonuniform grids than Δx on the uniform grids.

(b) 2D case. We solve the 2D linear advection equation

$$u_t + u_x + u_y = 0, \tag{29}$$

with the initial condition $u_0(x, y) = \sin(x + y)$ and periodic boundary conditions, on the computational domain $[0, 2\pi]^2$. The equation is solved on both the uniform grids and the nonuniform grids using these three different schemes. The final time $T = 1.0$. The L_1 and L_∞ errors, accuracy orders and computational times are reported in Table 5 and Table 6. On uniform grids, results in Table 5 show that the third-order accuracy

WENO3-MLP

N	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU time (s)
20	2.49528e-02		8.25621e-02		0.86
40	1.78005e-03	3.809	8.48559e-03	3.282	1.07
80	5.32923e-05	5.062	1.21674e-04	6.124	1.26
160	6.37260e-06	3.064	1.49104e-05	3.029	1.92
320	8.06644e-07	2.982	1.85535e-06	3.007	4.87
640	1.01473e-07	2.991	2.31763e-07	3.001	14.91

Classical WENO3

N	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU time (s)
20	3.19540e-02		8.25618e-02		0.09
40	8.37963e-03	1.931	3.34905e-02	1.302	0.32
80	2.15248e-03	1.961	1.30396e-02	1.361	1.19
160	4.61886e-04	2.220	4.64923e-03	1.488	4.61
320	5.94138e-05	2.959	1.00469e-03	2.210	18.10
640	3.88689e-06	3.934	8.33568e-05	3.591	71.69

WENO3 with KXRCF

N	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU time (s)
20	2.69500e-03		5.00773e-03		0.07
40	3.81686e-04	2.820	7.70200e-04	2.701	0.19
80	4.97924e-05	2.938	1.09005e-04	2.821	0.65
160	6.37260e-06	2.966	1.49104e-05	2.870	2.31
320	8.06644e-07	2.982	1.85535e-06	3.007	8.99
640	1.01473e-07	2.991	2.31763e-07	3.001	34.64

Table 4: Example 6.1.1, 1D linear advection equation with the initial condition $u_0 = \sin(x)$ on nonuniform grids with $c = 0.4$, and the smooth solution at $T = 1.0$. L_1 errors, L_∞ errors, numerical accuracy orders, and CPU time in seconds.

is achieved at very coarse meshes using the WENO3-MLP scheme, while using the classical WENO3 scheme, very refined meshes with $N = 320$ and $N = 640$ have to be used to reach the third-order accuracy for L_1 errors and L_∞ errors respectively. On the computational time of these schemes, we observe alike behavior as in the 1D case. Due to the model initialization, It takes a little more CPU times for the WENO3-MLP scheme than the other two schemes on relatively coarse meshes. As N becomes larger, the WENO3-MLP scheme is almost five times faster than the classical WENO3 scheme at $N = 640$. Comparing the numerical results of WENO3-MLP with those of WENO3 with the KXRCF indicator, we observe that both methods achieve the third-order accuracy on coarse meshes. Along with the mesh refinement, the L_1 and L_∞ numerical errors of WENO3-MLP and WENO3 with the KXRCF indicator are similar, and the WENO3-MLP scheme is more efficient in terms of computational time. Observations of the numerical results on the nonuniform grids are similar to these on the uniform grids. On the nonuniform grids with $c = 0.4$, the L_1 and L_∞ numerical errors, accuracy orders and CPU times of these three different schemes are reported in Table 6, which shows the advantages of WENO3-MLP scheme. Similar to the 1D case, the CPU times of the computations on the nonuniform grids for the same N and the same numerical scheme are longer than those on the uniform grids, since the time step sizes determined by the CFL condition are smaller on the nonuniform grids than those on the uniform grids.

Example 6.1.2 1D and 2D Burgers' equations

(a) 1D case We solve the 1D Burgers' equation

$$u_t + \left(\frac{u^2}{2}\right)_x = 0 \tag{30}$$

on both uniform and nonuniform grids. The initial condition is $u_0(x) = 1 + 0.5\sin(x)$, and the equation is solved by these three schemes on the domain $[0, 2\pi]$ with periodic boundary conditions. The final time $T = 1.0$ when the solution is still smooth. The L_1 and L_∞ numerical errors, accuracy orders and the CPU times of the three schemes on uniform grids are reported in Table 7. Similar to the example 6.1.1, we are able to obtain the third-order accuracy of the WENO3-MLP scheme for both the L_1 and the L_∞ numerical errors quickly on coarse meshes of these successively refined grids. However, for the classical WENO3 scheme, the third-order accuracy is reached at very refined meshes such as the grid with $N = 320$. Both the WENO3-MLP scheme and the WENO3 with KXRCF indicator accurately identify all cells as smooth on refined meshes, and their numerical errors are almost the same. On the computational costs, when the meshes are coarse, the computational speed of the WENO3-MLP scheme is slower than the other two schemes due to the MLP model initialization and some mis-identifications of the troubled-cells. However, on more refined meshes, the WENO3-MLP scheme is more efficient than the other two schemes. On the mesh with $N = 640$, it is more than four times faster than the classical WENO3 scheme. Note that the WENO3 scheme with KXRCF indicator is more expensive than the WENO3-MLP scheme on refined meshes due to the additional calculations of the indicator values κ_i

WENO3-MLP

N	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU time (s)
20	4.39767e-02		1.08863e-01		10.82
40	2.67914e-03	4.037	1.22072e-02	3.157	48.24
80	8.17696e-05	5.034	1.86980e-04	6.029	179.48
160	9.38821e-06	3.123	1.47366e-05	3.665	958.64
320	1.17233e-06	3.001	1.84148e-06	3.000	6858.05
640	1.46517e-07	3.000	2.30149e-07	3.000	54635.53

Classical WENO3

N	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU time (s)
20	5.51482e-02		1.08704e-01		8.14
40	1.44273e-02	1.935	4.45928e-02	1.286	63.44
80	3.53941e-03	2.027	1.67952e-02	1.409	515.02
160	7.01823e-04	2.334	5.55569e-03	1.596	4026.48
320	8.24321e-05	3.090	1.05262e-03	2.400	32454.42
640	5.22756e-06	3.979	7.42802e-05	3.825	258986.24

WENO3 with KXRCF

N	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU time (s)
20	4.78004e-03		7.44467e-03		3.09
40	6.00535e-04	2.993	9.41788e-04	2.983	18.45
80	7.50860e-05	3.000	1.17924e-04	2.998	124.35
160	9.38091e-06	3.001	1.47366e-05	3.000	924.00
320	1.17233e-06	3.000	1.84148e-06	3.000	7296.82
640	1.46517e-07	3.000	2.30149e-07	3.000	56497.32

Table 5: Example 6.1.1, 2D linear advection equation with the initial condition $u_0 = \sin(x+y)$ on uniform grids, and the smooth solution at $T = 1.0$. L_1 errors, L_∞ errors, numerical accuracy orders, and CPU time in seconds.

WENO3-MLP

N	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU time (s)
20	4.61634e-02		1.14949e-01		34.52
40	3.50861e-03	3.718	1.83381e-02	2.648	110.15
80	1.03974e-04	5.077	4.45698e-04	5.363	364.67
160	1.13823e-05	3.191	3.11577e-05	3.838	1774.46
320	1.44893e-06	2.974	4.18558e-06	2.896	13225.04
640	1.82860e-07	2.986	5.48604e-07	2.932	101508.27

Classical WENO3

N	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU time (s)
20	5.72797e-02		1.23107e-01		18.03
40	1.64631e-02	1.799	5.35985e-02	1.120	137.84
80	4.18887e-03	1.975	2.25882e-02	1.247	1088.37
160	9.25559e-04	2.178	8.32115e-03	1.441	8825.30
320	1.32770e-04	2.801	2.25021e-03	1.887	71086.71
640	9.66571e-06	3.780	2.80673e-04	3.003	565822.64

WENO3 with KXRCF

N	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU time (s)
20	4.44972e-03		8.99429e-03		6.67
40	6.36454e-04	2.806	1.33311e-03	2.754	36.58
80	8.70057e-05	2.871	2.33311e-04	2.514	243.78
160	1.13520e-05	2.938	3.11550e-05	2.905	1871.81
320	1.44893e-06	2.970	4.18558e-06	2.896	13353.93
640	1.82860e-07	2.986	5.48604e-07	2.932	105338.73

Table 6: Example 6.1.1, 2D linear advection equation with the initial condition $u_0 = \sin(x+y)$ on nonuniform grids with $c = 0.4$, and the smooth solution at $T = 1.0$. L_1 errors, L_∞ errors, numerical accuracy orders, and CPU time in seconds.

WENO3-MLP					
N	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU time (s)
20	1.64964e-02		5.74768e-02		0.83
40	1.00734e-03	4.034	6.20006e-03	3.213	0.86
80	6.81860e-05	3.885	6.14000e-04	3.336	1.01
160	8.61419e-06	2.985	7.91393e-05	2.956	1.51
320	1.07392e-06	3.004	9.89504e-06	3.000	3.30
640	1.34065e-07	3.002	1.23293e-06	3.005	9.78

Classical WENO3					
N	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU time (s)
20	2.04476e-02		6.12192e-02		0.06
40	5.59372e-03	1.870	1.81828e-02	1.751	0.19
80	1.29446e-03	2.111	6.73150e-03	1.434	0.73
160	2.07298e-04	2.643	1.74755e-03	1.946	2.77
320	1.61104e-05	3.686	1.90709e-04	3.196	10.81
640	1.00917e-06	3.997	8.02618e-06	4.571	43.65

WENO3 with KXRCF					
N	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU time (s)
20	3.23937e-03		2.21190e-02		0.04
40	5.10365e-04	2.666	4.02810e-03	2.457	0.14
80	6.81860e-05	2.904	6.14000e-04	2.714	0.49
160	8.61419e-06	2.985	7.91393e-05	2.956	1.75
320	1.07392e-06	3.004	9.89504e-06	3.000	6.73
640	1.34065e-07	3.002	1.23293e-06	3.005	26.63

Table 7: Example 6.1.2, 1D Burgers equation with the initial condition $u_0 = 1 + 0.5\sin(x)$ on uniform grids, and the smooth solution at $T = 1.0$. L_1 errors, L_∞ errors, numerical accuracy orders, and CPU time in seconds.

on all cells. On the mesh with $N = 640$, the WENO3-MLP scheme is more than two times faster than the WENO3 scheme with KXRCF indicator. Similar to the example 6.1.1, we also report the numerical results on uniform grids using the WENO3-MLP scheme without the additional WENO3-MLP reconstruction for updated cell boundary values $u^-_{i+1/2}, u^+_{i-1/2}$ in the neural network input data. Table 8 shows that without this additional WENO3-MLP reconstruction, the numerical errors are much larger than the WENO3-MLP scheme' errors in Table 7, and the desired third-order accuracy is achieved only when the meshes are very refined. Similar to the linear equation example, the CPU times on all of the grids are larger than those reported in Table 7 for the WENO3-MLP, because more cells are mis-identified as troubled-cells on which the more costly WENO3 reconstruction scheme is used instead of the linear scheme. Again, it is verified in this nonlinear equation example that the additional WENO3-

N	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU time (s)
20	1.83320e-02		6.11408e-02		0.86
40	3.14252e-03	2.544	1.42803e-02	2.098	0.94
80	5.20881e-04	2.593	4.61958e-03	1.628	1.13
160	6.95087e-05	2.906	1.05572e-03	2.130	1.86
320	5.52322e-06	3.654	9.52469e-05	3.470	4.61
640	4.36298e-07	3.662	6.97639e-06	3.771	14.85

Table 8: Example 6.1.2, 1D Burgers' equation with the initial condition $u_0 = 1 + 0.5\sin(x)$ on uniform grids, and the smooth solution at $T = 1.0$. WENO3-MLP scheme without the additional WENO3-MLP reconstruction for updated cell boundary values $u^-_{i+1/2}, u^+_{i-1/2}$ in the neural network input data. L_1 errors, L_∞ errors, numerical accuracy orders, and CPU time in seconds.

MLP reconstruction for obtaining updated cell boundary values in the neural network input data improves the accuracy of the MLP model for indication of the troubled-cells, and actually makes the whole WENO3-MLP scheme more efficient. On the nonuniform grids with $c = 0.4$, the behaviors of these schemes are similar to these on the uniform grids. Table 9 shows the results of the accuracy, convergence orders, and computational costs of these three schemes. It is verified that the WENO3-MLP scheme achieves the desired third-order accuracy quickly in the mesh refinement process, and needs less computational time than the other two methods on relatively refined meshes. It is also noted that the computational times are longer on the nonuniform grids than those on the uniform grids for the same N and the same numerical scheme, since the time step sizes determined by the CFL condition are smaller on the nonuniform grids.

(b) 2D case We solve the 2D Burgers' equation

$$u_t + \left(\frac{u^2}{2}\right)_x + \left(\frac{u^2}{2}\right)_y = 0 \tag{31}$$

on both uniform and nonuniform grids. The initial condition is $u_0(x, y) = 1+0.5\sin(x+y)$, and the domain is $[0, 2\pi]^2$ with periodic boundary conditions. The final time $T = 0.3$ when the solution is still smooth. The L_1 and L_∞ numerical errors, accuracy orders and computational times are reported in Table 10 for these three schemes on uniform grids. We observe the third-order accuracy is reached quickly in the mesh refinement process for the WENO3-MLP scheme, while much more refined mesh such as $N = 320$ has to be used to achieve that for the classical WENO3 scheme. Similar to the 1D case, the WENO3-MLP scheme mis-identified some smooth cells as troubled-cells on coarse meshes such as $N = 20$, so with the addition cost for the neural network initialization, the computational cost is slightly larger than the classical WENO3 scheme on the grid with $N = 20$. However, the WENO3-MLP scheme needs much less CPU time than the classical WENO3 scheme when the meshes are refined. On the mesh with $N = 640$, it is almost five times faster than the classical WENO3 scheme. Also, similar observations

WENO3-MLP					
N	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU time (s)
20	1.86556e-02		4.86616e-02		0.96
40	1.29499e-03	3.849	1.06354e-02	2.194	1.01
80	8.71291e-05	3.894	8.80623e-04	3.594	1.34
160	1.12077e-05	2.959	1.20987e-04	2.864	2.57
320	1.40668e-06	2.994	1.53541e-05	2.978	6.71
640	1.76081e-07	2.998	1.92869e-06	2.993	22.37
Classical WENO3					
N	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU time (s)
20	2.25577e-02		5.22689e-02		0.13
40	6.63330e-03	1.766	2.13834e-02	1.289	0.46
80	1.51265e-03	2.133	8.49442e-03	1.332	1.74
160	2.68920e-04	2.492	2.50535e-03	1.762	6.84
320	2.50854e-05	3.422	3.60731e-04	2.796	27.12
640	1.61514e-06	3.957	1.83464e-05	4.297	107.16
WENO3 with KXRCF					
N	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU time (s)
20	3.90191e-03		2.62982e-02		0.11
40	6.15443e-04	2.664	5.77837e-03	2.186	0.34
80	8.71291e-05	2.820	8.80623e-04	2.714	1.20
160	1.12077e-05	2.959	1.20987e-04	2.864	4.42
320	1.40668e-06	2.994	1.53541e-05	2.978	16.72
640	1.76081e-07	2.998	1.92869e-06	2.993	64.72

Table 9: Example 6.1.2, 1D Burgers' equation with the initial condition $u_0 = 1 + 0.5\sin(x)$ on nonuniform grids with $c = 0.4$, and the smooth solution at $T = 1.0$. L_1 errors, L_∞ errors, numerical accuracy orders, and CPU time in seconds.

WENO3-MLP					
N	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU time (s)
20	1.49964e-02		5.78828e-02		4.76
40	6.13649e-04	4.611	2.79462e-03	4.372	12.35
80	4.18358e-05	3.875	1.87973e-04	3.894	55.39
160	5.22144e-06	3.002	2.35723e-05	2.995	407.94
320	6.52136e-07	3.001	2.93904e-06	3.004	3052.25
640	8.14606e-08	3.001	3.66911e-07	3.002	24099.23
Classical WENO3					
N	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU time (s)
20	1.87090e-02		6.06548e-02		3.82
40	4.14211e-03	2.175	1.57149e-02	1.948	29.49
80	8.82609e-04	2.231	4.90531e-03	1.680	226.09
160	1.38003e-04	2.677	1.23876e-03	1.985	1801.25
320	1.06056e-05	3.701	1.20669e-04	3.360	14828.44
640	7.48933e-07	3.824	4.82966e-06	4.643	118288.89
WENO3 with KXRCF					
N	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU time (s)
20	2.58898e-03		1.03698e-02		1.39
40	3.35150e-04	2.950	1.48462e-03	2.804	8.57
80	4.18358e-05	3.002	1.87973e-04	2.981	55.39
160	5.22144e-06	3.002	2.35723e-05	2.995	404.55
320	6.52136e-07	3.001	2.93904e-06	3.004	3164.50
640	8.14606e-08	3.001	3.66911e-07	3.002	24498.46

Table 10: Example 6.1.2, 2D Burgers' equation with the initial condition $u_0 = 1+0.5\sin(x+y)$ on uniform grids, and the smooth solution at $T = 0.3$. L_1 errors, L_∞ errors, numerical accuracy orders, and CPU time in seconds.

as in the 1D Burgers equation are obtained for a comparison between the results of the WENO3-MLP scheme and the results of the WENO3 scheme with KXRCF indicator. The numerical errors of the WENO3 scheme with KXRCF indicator are smaller when the meshes are coarse, since the KXRCF indicator correctly identifies all smooth cells. When the meshes are refined, the numerical errors of these two methods are identical, and the WENO3-MLP scheme is slightly faster on the refined meshes with $N = 320$ and $N = 640$. On the nonuniform grids with $c = 0.4$, the L_1 and L_∞ numerical errors, accuracy orders and CPU times of these three different schemes are reported in Table 11, which shows the similar advantages of WENO3-MLP scheme as the observations on uniform grids. Also similar to the 1D case, the computational times of the 2D case on the nonuniform grids are longer than those on the uniform grids for the same N and the same numerical scheme, because the time step sizes which are determined by the CFL condition are smaller on these nonuniform grids.

WENO3-MLP

N	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU time (s)
20	1.68199e-02		6.82497e-02		13.69
40	9.42874e-04	4.157	1.07975e-02	2.660	26.15
80	5.65633e-05	4.059	5.14232e-04	4.392	107.48
160	7.43579e-06	2.927	7.39962e-05	2.797	760.88
320	9.50476e-07	2.968	9.35186e-06	2.984	5864.76
640	1.20083e-07	2.985	1.22723e-06	2.930	45548.12

Classical WENO3

N	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU time (s)
20	1.97179e-02		6.98374e-02		8.29
40	4.76419e-03	2.049	2.13162e-02	1.712	60.79
80	1.09048e-03	2.127	7.52218e-03	1.503	497.33
160	1.96522e-04	2.472	2.67654e-03	1.491	3908.38
320	1.97392e-05	3.316	5.22376e-04	2.357	31739.28
640	1.42865e-06	3.788	3.41968e-05	3.933	257969.77

WENO3 with KXRCF

N	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU time (s)
20	2.55798e-03		1.32908e-02		3.01
40	3.94084e-04	2.698	2.51690e-03	2.401	16.83
80	5.65035e-05	2.802	5.14233e-04	2.291	110.49
160	7.43579e-06	2.926	7.39962e-05	2.797	812.58
320	9.50476e-07	2.968	9.35186e-06	2.984	6051.74
640	1.20083e-07	2.985	1.22723e-06	2.930	48429.11

Table 11: Example 6.1.2, 2D Burgers' equation with the initial condition $u_0 = 1+0.5\sin(x+y)$ on nonuniform grids with $c = 0.4$, and the smooth solution at $T = 0.3$. L_1 errors, L_∞ errors, numerical accuracy orders, and CPU time in seconds.

Example 6.1.3 1D Euler system of equations We solve the one-dimensional Euler system of equations

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho \\ \rho u \\ E\end{pmatrix} + \frac{\partial}{\partial x}\begin{pmatrix}\rho u \\ \rho u^2 + p \\ u(E+p)\end{pmatrix} = 0, \tag{32}$$

where ρ is the density, u is the velocity, p is the pressure and $E = p/(\gamma - 1) + \frac{1}{2}\rho u^2$ is the total energy. γ is the ratio of specific heat, with $\gamma = 1.4$ for air. Consider the computational domain $[0, 2\pi]$ with periodic boundary conditions, and the initial conditions $\rho_0 = 1 + 0.2\sin(x)$, $u_0 = 1$, and $p_0 = 1$. The exact solution of the density is $\rho(x,t) = 1 + 0.2\sin(x - t)$. These three schemes (WENO3-MLP, classical WENO3, and WENO3 with KXRCF indicator) are applied for solving the equations till the final time $T = 2.0$ on both uniform and nonuniform grids. We report the numerical results on uniform grids, including the L_1 and L_∞ errors, numerical accuracy orders of these three schemes for the density ρ, and their CPU times in Table 12. It is observed that the WENO3-MLP scheme yields identical numerical errors as the WENO3 scheme with KXRCF indicator, since both methods have correctly identified all smooth cells. Both schemes achieve the third-order accuracy quickly on coarse meshes of these successively refined grids, and much smaller L_1 and L_∞ errors than the classical WENO3 scheme. Comparing the CPU times of the classical WENO3 scheme and the WENO3-MLP scheme, we see similar behavior as in the scalar-equation cases. On the coarsest mesh with $N = 20$, due to the neural network model's initialization, the WENO3-MLP scheme takes a little more CPU times than the classical WENO3 scheme does. However, when the meshes are refined, the WENO3-MLP scheme is much faster. On the grid with $N = 320$, the WENO3-MLP scheme is more than four times faster than the classical WENO3 scheme. On the grid with $N = 640$, the WENO3-MLP scheme is almost five times faster. Similarly, although the WENO3 scheme with KXRCF indicator is faster than the WENO3-MLP scheme on coarse meshes with $N = 20$ and $N = 40$, it is slower than the WENO3-MLP scheme on more refined meshes. On the grid with $N = 640$, the WENO3-MLP scheme is more than two times faster than the WENO3 scheme with KXRCF indicator. On the nonuniform grids with $c = 0.4$, the numerical results are shown in Table 13, which includes the L_1 and L_∞ numerical errors, numerical accuracy orders of these three schemes for the density ρ, and CPU times. We observe similar performances of the three schemes on the nonuniform grids to those on the uniform grids for this example. Also, similar to the scalar-equation examples, the computational times on the nonuniform grids are longer than those on the uniform grids for the same N and the same numerical scheme because of the smaller time step sizes for simulations on the nonuniform grids.

Example 6.1.4 2D Euler system of equations We solve the two-dimensional Euler system of equations:

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho \\ \rho u \\ \rho v \\ E\end{pmatrix} + \frac{\partial}{\partial x}\begin{pmatrix}\rho u \\ \rho u^2 + p \\ \rho uv \\ u(E+p)\end{pmatrix} + \frac{\partial}{\partial y}\begin{pmatrix}\rho v \\ \rho uv \\ \rho v^2 + p \\ v(E+p)\end{pmatrix} = 0, \tag{33}$$

WENO3-MLP					
N	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU time (s)
20	2.11336e-04		3.79572e-04		0.76
40	2.43791e-05	3.116	4.58692e-05	3.049	1.32
80	2.98225e-06	3.031	5.76940e-06	2.991	2.84
160	3.70701e-07	3.008	7.23836e-07	2.995	7.58
320	4.62495e-08	3.003	9.06858e-08	2.997	23.83
640	5.77707e-09	3.001	1.13486e-08	2.998	83.30
Classical WENO3					
N	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU time (s)
20	1.08910e-02		2.45275e-02		0.53
40	4.08327e-03	1.415	9.43978e-03	1.378	1.85
80	1.04787e-03	1.962	3.08868e-03	1.612	6.60
160	4.26152e-05	4.620	2.48180e-04	3.638	25.99
320	1.40605e-06	4.922	1.08241e-05	4.519	100.06
640	7.82800e-08	4.167	3.38345e-07	5.000	404.75
WENO3 with KXRCF					
N	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU time (s)
20	2.11336e-04		3.79572e-04		0.36
40	2.43791e-05	3.116	4.58692e-05	3.049	1.04
80	2.98225e-06	3.031	5.76940e-06	2.991	3.37
160	3.70701e-07	3.008	7.23836e-07	2.995	11.91
320	4.62495e-08	3.003	9.06858e-08	2.997	45.25
640	5.77707e-09	3.001	1.13486e-08	2.998	172.46

Table 12: Example 6.1.3, 1D Euler system of equations with the initial condition $\rho_0 = 1 + 0.2\sin(x)$, $u_0 = 1$, and $p_0 = 1$ on uniform grids, and the smooth solution at $T = 2.0$. L_1 errors, L_∞ errors, numerical accuracy orders of three schemes for the density ρ, and CPU times in seconds.

WENO3-MLP

N	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU time (s)
20	4.26307e-04		7.37636e-04		1.33
40	6.02796e-05	2.822	1.23472e-04	2.579	2.64
80	8.04390e-06	2.906	1.74796e-05	2.820	6.29
160	1.04291e-06	2.947	2.24846e-06	2.959	17.23
320	1.32627e-07	2.975	3.01634e-07	2.898	54.07
640	1.67135e-08	2.988	3.80080e-08	2.988	195.10

Classical WENO3

N	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU time (s)
20	1.30306e-02		3.01462e-02		1.27
40	4.11116e-03	1.664	1.23309e-02	1.290	4.50
80	1.09660e-03	1.907	3.95008e-03	1.642	16.65
160	1.16034e-04	3.240	5.41672e-04	2.866	65.25
320	2.50577e-06	5.533	1.98510e-05	4.770	249.93
640	1.55730e-07	4.008	7.08836e-07	4.808	1000.67

WENO3 with KXRCF

N	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU time (s)
20	4.26307e-04		7.37636e-04		0.84
40	6.02796e-05	2.822	1.23472e-04	2.579	2.49
80	8.04390e-06	2.906	1.74796e-05	2.820	8.04
160	1.04291e-06	2.947	2.24846e-06	2.959	28.90
320	1.32627e-07	2.975	3.01634e-07	2.898	106.64
640	1.67135e-08	2.988	3.80080e-08	2.988	405.42

Table 13: Example 6.1.3, 1D Euler equations with the initial condition $\rho_0 = 1 + 0.2\sin(x)$, $u_0 = 1$, and $p_0 = 1$ on nonuniform grids with $c = 0.4$, and the smooth solution at $T = 2.0$. L_1 errors, L_∞ errors, numerical accuracy orders of three schemes for the density ρ, and CPU times in seconds.

where ρ is the density, u is the x-directional velocity, v is the y-directional velocity, p is the pressure and $E = p/(\gamma-1) + \frac{1}{2}\rho(u^2+v^2)$ is the total energy. γ is the ratio of specific heat, and $\gamma = 1.4$ for air. Consider the computational domain $[0, 2\pi]^2$ with periodic boundary conditions, and the initial condition $\rho_0 = 1+0.2\sin(x+y)$, $u_0 = 0.7$, $v_0 = 0.3$, and $p_0 = 1$. The exact solution of density is $\rho(x, y, t) = 1+0.2\sin(x+y-2t)$. We solve the system of equations with the final time $T = 0.3$ on both uniform and nonuniform grids. Table 14 shows the L_1 and L_∞ numerical errors, numerical accuracy orders of the WENO3-MLP scheme, the classical WENO3 scheme, and the WENO3 scheme with KXRCF indicator on the uniform grids for the density ρ, and their CPU times. Again, the WENO3-MLP scheme yields much smaller numerical errors than those of the classical WENO3 scheme, and has similar numerical errors to those of the WENO3 scheme with KXRCF indicator. It is found that the neural network model in the WENO3-MLP scheme correctly identifies all smooth cells on the grids with $N = 40$ or larger, which leads the WENO3-MLP scheme to be able to reach the desired 3rd-order accuracy quickly during the mesh refinement. Furthermore, the WENO3-MLP scheme needs much less computational time than the classical WENO3 scheme on all grids. For example, on the grid with $N = 640$, the WENO3-MLP scheme is about five times faster than the classical WENO3 scheme. It is also faster than the WENO3 scheme with KXRCF indicator on the grids with $N = 40$ or larger. On the nonuniform grids with $c = 0.4$, the numerical results are reported in Table 15, from which the similar observations to those on the uniform grids are obtained for these three numerical schemes.

6.2 Numerical examples with nonsmooth solutions

Example 6.2.1 1D linear advection equation We consider the 1D linear advection equation (28) with nonsmooth solution. The equation is solved on both uniform and nonuniform grids with the discontinuous initial condition

$$u_0 = \begin{cases} 10(x-0.2), & \text{if } 0.2 < x \leq 0.3 \\ 10(0.4-x), & \text{if } 0.3 < x < 0.4 \\ 1, & \text{if } 0.6 < x < 0.8 \\ 100(x-1)(1.2-x), & \text{if } 1.0 < x < 1.2 \\ 0, & \text{otherwise.} \end{cases} \tag{34}$$

The computational domain is $[0, 1.4]$, and periodic boundary conditions are applied. The equation is solved till the final time $T = 1.4$. Fig. 3 presents the numerical solutions of the classical WENO3 scheme, the WENO3-MLP scheme, and the WENO3 scheme with KXRCF indicator on both the uniform grid and the nonuniform grid with the number of computational cells $N = 160$. The reference solution is provided using the exact solution of the equation. In this example, the size of buffer zones is taken as $N_b = 7$ and $N_{b,\kappa} = 7$ for the WENO3-MLP scheme and the WENO3 scheme with KXRCF indicator respectively, which are increased from the default value $N_b = N_{b,\kappa} = 3$ in the other examples, because the numerical experiment shows that the larger buffer zones have a better coverage of the transition regions near discontinuities here and make the WENO3-MLP scheme more stable. This observation is consistent

WENO3-MLP					
N	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU time (s)
20	1.05886e-03		1.58511e-03		8.36
40	9.56576e-05	3.468	1.50791e-04	3.394	44.46
80	1.19536e-05	3.000	1.88063e-05	3.003	287.08
160	1.49177e-06	3.002	2.34882e-06	3.001	2083.33
320	1.86300e-07	3.001	2.93386e-07	3.001	15719.07
640	2.32770e-08	3.001	3.66569e-08	3.001	125350.15
Classical WENO3					
N	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU time (s)
20	6.80181e-03		1.64538e-02		19.78
40	1.55986e-03	2.125	5.25255e-03	1.647	156.24
80	2.92839e-04	2.413	1.57353e-03	1.739	1212.33
160	2.75379e-05	3.411	2.16386e-04	2.862	9844.85
320	1.91191e-06	3.848	1.00506e-05	4.428	78487.71
640	1.23234e-07	3.956	3.60289e-07	4.802	621617.04
WENO3 with KXRCF					
N	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU time (s)
20	7.66879e-04		1.19564e-03		7.62
40	9.56576e-05	3.003	1.50791e-04	2.987	46.65
80	1.19536e-05	3.000	1.88063e-05	3.003	306.46
160	1.49177e-06	3.002	2.34882e-06	3.001	2215.55
320	1.86300e-07	3.001	2.93386e-07	3.001	16763.42
640	2.32770e-08	3.001	3.66569e-08	3.001	131254.76

Table 14: Example 6.1.4, 2D Euler system of equations with the initial condition $\rho_0 = 1 + 0.2\sin(x+y)$, $u_0 = 0.7$, $v_0 = 0.3$, and $p_0 = 1$ on uniform grids, and the smooth solution at $T = 0.3$. L_1 errors, L_∞ errors, numerical accuracy orders of three schemes for the density ρ, and CPU time in seconds.

WENO3-MLP					
N	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU time (s)
20	8.90705e-04		1.74243e-03		17.25
40	1.10518e-04	3.011	2.95386e-04	2.560	86.27
80	1.56916e-05	2.816	5.00372e-05	2.562	562.01
160	2.10862e-06	2.896	7.08265e-06	2.821	4047.49
320	2.72805e-07	2.950	9.79135e-07	2.855	30823.17
640	3.46733e-08	2.976	1.33509e-07	2.875	241026.07
Classical WENO3					
N	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU time (s)
20	7.75540e-03		2.34874e-02		45.77
40	1.82499e-03	2.087	7.58974e-03	1.630	331.64
80	3.73134e-04	2.290	2.39292e-03	1.665	2680.40
160	4.69420e-05	2.991	6.04227e-04	1.986	21335.99
320	3.48530e-06	3.752	5.25303e-05	3.524	173557.89
640	2.38769e-07	3.868	2.28757e-06	4.521	1391268.17
WENO3 with KXRCF					
N	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU time (s)
20	7.50626e-04		1.79105e-03		16.56
40	1.10518e-04	2.764	2.95386e-04	2.600	89.42
80	1.56916e-05	2.816	5.00372e-05	2.562	622.69
160	2.10862e-06	2.896	7.08265e-06	2.821	4288.19
320	2.72805e-07	2.950	9.79135e-07	2.855	33048.76
640	3.46733e-08	2.976	1.33509e-07	2.875	264312.77

Table 15: Example 6.1.4, 2D Euler system of equations with the initial condition $\rho_0 = 1 + 0.2\sin(x+y)$, $u_0 = 0.7$, $v_0 = 0.3$, and $p_0 = 1$ on nonuniform grids with $c = 0.4$, and the smooth solution at $T = 0.3$. L_1 errors, L_∞ errors, numerical accuracy orders of three schemes for the density ρ, and CPU time in seconds.

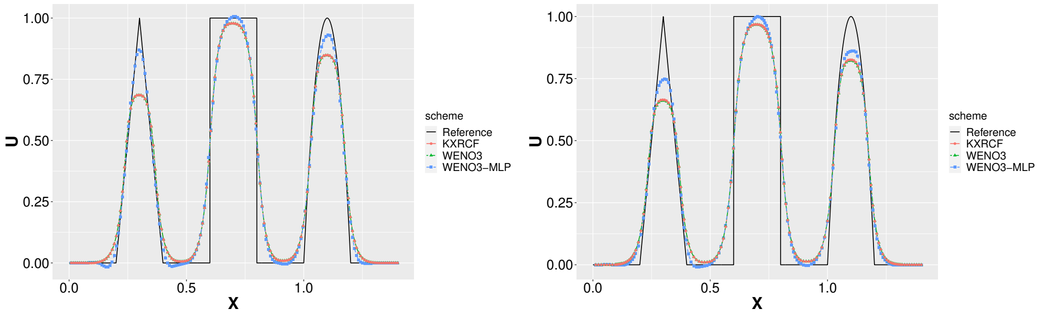

Figure 3: Example 6.2.1, numerical solutions of three schemes for the 1D linear equation with discontinuous initial condition. The time $T = 1.4$. Number of computational cells $N = 160$. Left: uniform grid; right: nonuniform grid with $c = 0.4$.

with the results reported in the hybrid finite difference scheme [49]. Fig. 3 shows that the WENO3-MLP scheme has a better resolution than the other two schemes in this example, although there are some slight under-shoots in its solution, especially on the uniform grid.

We compare the troubled-cells marked by the WENO3-MLP scheme and the WENO3 with KXRCF indicator on the nonuniform grid at different times $t = 0.05$, $t = 0.1$, $t = 0.3$, and $t = 1.4$. The results are reported in Fig. 4, which shows that the non-smooth regions of the solution are captured well by both schemes in the time evolutions. In the figure, the values of $1 - P$ of all computational cells are shown for the WENO3-MLP scheme, where the values of P are the outputs of the MLP model. A dashed cutoff line $1 -$ threshold of $P = 0.1$ is included in the plots. So if the $1 - P$ value of a cell is above the dashed cutoff line, this cell is indicated to be a troubled-cell. Similarly, for the convenience of comparison, the indicator values κ_i of the WENO3 scheme with KXRCF indicator are re-scaled by a factor of 10 in the figure, and a dashed cutoff line $1/10 = 0.1$ is also included in the plots. So if the $\kappa_i/10$ value of a cell is above the dashed cutoff line, this cell is indicated to be a troubled-cell. Fig. 4 clearly shows that the MLP model flags much less troubled-cells than the KXRCF indicator, which makes the hybrid WENO3-MLP scheme less dissipative, sharper in resolution and more efficient than the hybrid WENO3 with KXRCF indicator, although more dissipations in the hybrid WENO3 with KXRCF indicator provide stronger stability.

Example 6.2.2 1D and 2D Burgers' equations

(a) 1D case We solve the 1D Burgers' equation (30) with the initial condition $u_0(x) = 1 + 0.5\sin(x)$ and periodic boundary conditions on the domain $[0, 2\pi]$. Both the uniform grid and the nonuniform grid are used, and the equation is solved till the final time $T = 6.0$ when the shock wave has formed in the solution. Comparisons of numerical solutions computed by the classical WENO3, the WENO3-MLP, and the WENO3 with KXRCF indicator on the uniform grid and the nonuniform grid with $c = 0.4$ are shown in Fig. 5. Three numerical methods properly capture the

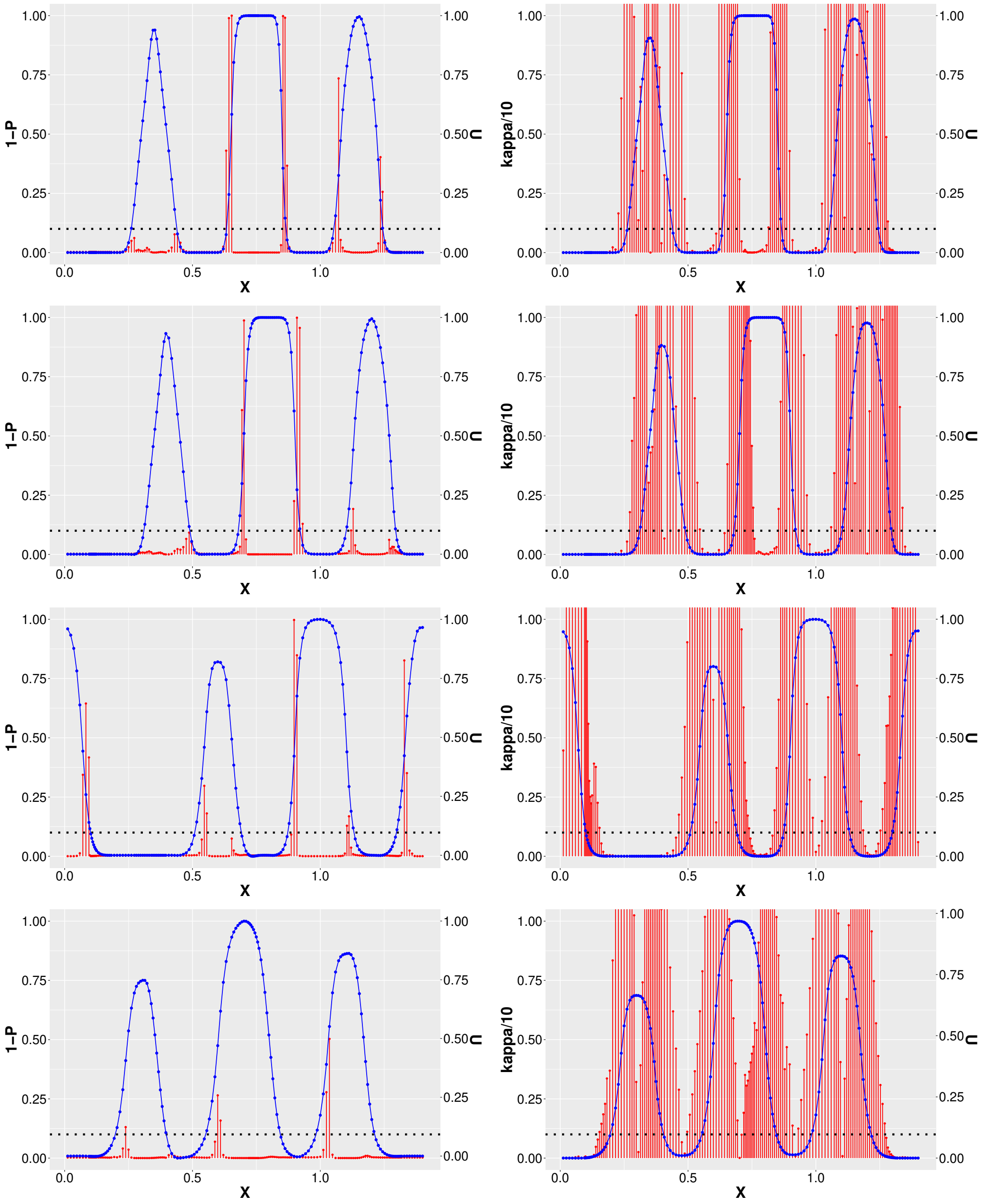

Figure 4: Example 6.2.1, numerical solutions of the 1D linear equation with nonsmooth solution and the troubled-cells (i.e., the indicator values are above the dashed cutoff line) marked by WENO3-MLP and WENO3 with KXRCF indicator on the nonuniform grid with $c = 0.4$ and the number of cells $N = 160$. Blue: numerical solutions; red: corresponding indicator values for all cells. Left column: WENO3-MLP; right column: WENO3 with KXRCF indicator; first row: $t = 0.05$; second row: $t = 0.1$; third row: $t = 0.3$; fourth row: $t = 1.4$.

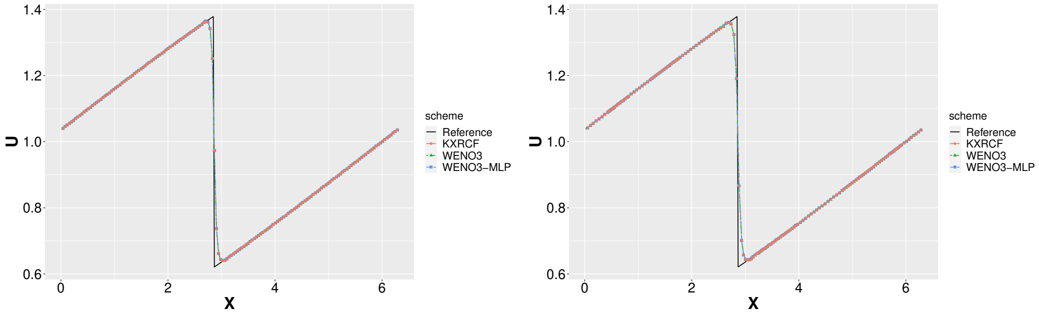

Figure 5: Example 6.2.2, numerical solutions of the 1D Burgers' equation with the initial condition $u_0 = 1 + 0.5\sin(x)$ at $T = 6.0$. Number of computational cells $N = 160$. Left: uniform grid; right: nonuniform grid with $c = 0.4$.

shock wave and yield similar results in this example on both the uniform grid and the nonuniform grid. We also compare the troubled-cells marked by the WENO3-MLP scheme and the WENO3 with KXRCF indicator. The results on the nonuniform grid with $c = 0.4$ at the final time $T = 6.0$ are reported in Fig. 6. Similar to the previous example, the KXRCF indicator marks more cells as troubled-cells than the neural network MLP model, while both hybrid schemes capture the discontinuity location well and are nonlinear stable.

(b) 2D case We solve the 2D Burgers' equation (31) with the initial condition $u_0(x, y) = 1 + 0.5\sin(x + y)$ and periodic boundary conditions on the domain $[0, 2\pi]^2$. Both the uniform grid and the nonuniform grid are used, and the equation is solved till the final time $T = 6.0$ when the shock waves have developed in the solution. In Fig. 7, we present the numerical solutions of the classical WENO3, the WENO3-MLP, and the WENO3 with KXRCF indicator on the grid with number of cells 160×160 in the 1D diagonal line $x = y$ of the domain. While the classical WENO3 scheme and the hybrid WENO3 with KXRCF indicator provide almost identical results, we observe that the numerical solution of the WENO3-MLP scheme is slightly closer to the exact solution near the shock transition location on both the uniform grid and the nonuniform grid with $c = 0.4$. The results of comparing the troubled-cells marked by the neural network indicator and the KXRCF indicator on the nonuniform grid are reported in Fig. 8. The consistent observation with the 1D case is observed. Much more troubled-cells are marked by the KXRCF indicator than the WENO3-MLP scheme.

Example 6.2.3 1D Buckley-Leverett equation We consider the 1D Buckley-Leverett equation

$$u_t + \left(\frac{4u^2}{4u^2 + (1-u)^2}\right)_x = 0, \tag{35}$$

where the flux is non-convex. The computational domain is $[-1, 1]$, and periodic boundary conditions are applied. The initial condition is $u_0 = 1$ in $[-0.5, 0]$, and $u_0 = 0$

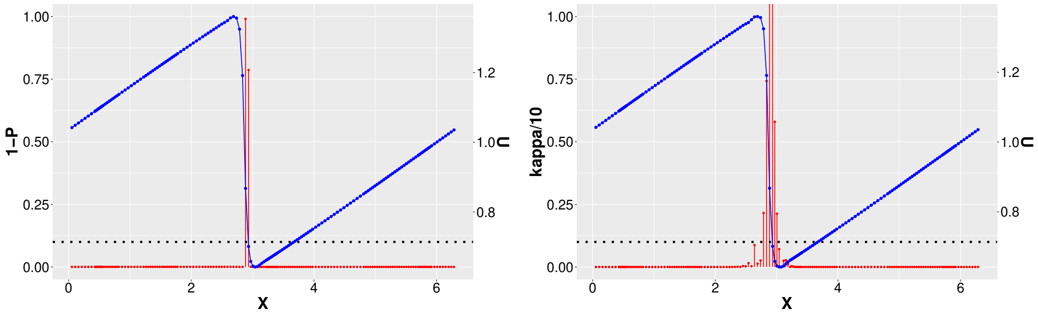

Figure 6: Example 6.2.2, numerical solutions of the 1D Burgers' equation and the troubled-cells (i.e., the indicator values are above the dashed cutoff line) marked by WENO3-MLP and WENO3 with KXRCF indicator on the nonuniform grid with $c = 0.4$ at the time $T = 6.0$. Number of computational cells $N = 160$. Blue: numerical solutions; red: corresponding indicator values for all cells. Left: WENO3-MLP; right: WENO3 with KXRCF indicator.

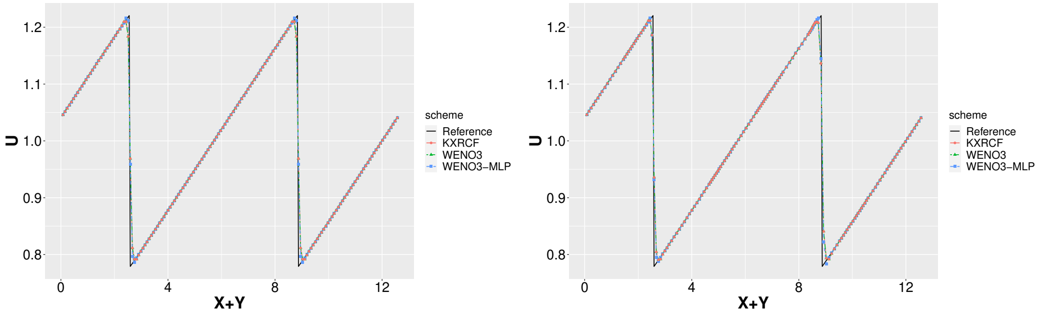

Figure 7: Example 6.2.2, numerical solutions of the 2D Burgers' equation with the initial condition $u_0 = 1 + 0.5\sin(x+y)$ at $T = 6.0$. 1D-cut plot in the $x = y$ diagonal line. Number of cells $N \times N = 160 \times 160$. Left: uniform grid; right: nonuniform grid with $c = 0.4$.

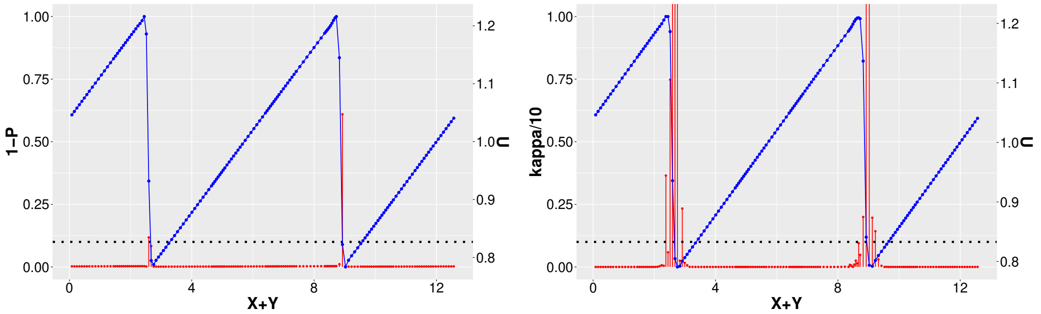

Figure 8: Example 6.2.2, numerical solutions of the 2D Burgers' equation and the troubled-cells (i.e., their indicator values are above the dashed cutoff line) marked by the WENO3-MLP and the WENO3 with KXRCF indicator on the nonuniform grid with $c = 0.4$ at the time $T = 6.0$. Number of cells $N \times N = 160 \times 160$. Blue: numerical solutions; red: corresponding indicator values for all cells. Left: WENO3-MLP; right: WENO3 with KXRCF indicator.

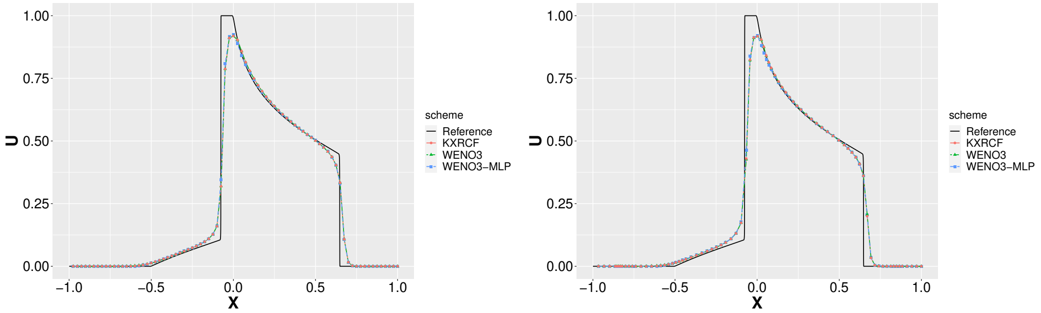

Figure 9: Example 6.2.3, numerical solutions of 1D Buckley-Leverett equation at $T = 0.4$. Number of cells $N = 80$. Left: on uniform grid; right: on nonuniform grid with $c = 0.4$.

elsewhere. We compute the solution till the final time $T = 0.4$ with $N = 80$ on both the uniform grid and the nonuniform grid with $c = 0.4$. The solution of this example has rarefaction wave, shock wave and contact discontinuity. We present the comparisons of the numerical solutions of the WENO3-MLP scheme, the classical WENO3 scheme and the WENO3 with KXRCF indicator scheme in Fig. 9. It shows that the numerical solutions of all three schemes agree well and are comparable. Here the reference solution is calculated using the classical WENO3 scheme on a very refined grid with $N = 5120$. In Fig. 10, we compare the number of troubled-cells marked by the WENO3-MLP scheme and the WENO3 with KXRCF indicator scheme. Again, the WENO3 with KXRCF indicator marks more cells as troubled-cells than the neural network MLP model. Both hybrid WENO3 schemes capture the discontinuity locations well and are nonlinear stable for this problem with a non-convex flux.

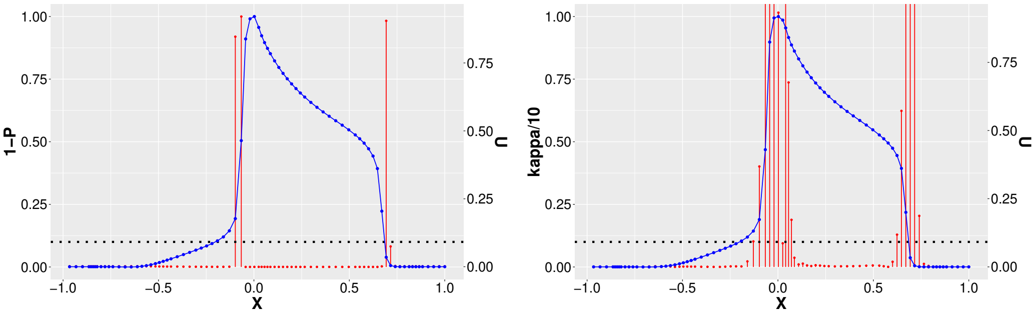

Figure 10: Example 6.2.3, numerical solutions of the 1D Buckley-Leverett equation and the troubled-cells (i.e., the indicator values are above the dashed cutoff line) marked by WENO3-MLP and WENO3 with KXRCF indicator on the nonuniform grid with $c = 0.4$ at $T = 0.4$. Number of computational cells $N = 80$. Blue: numerical solutions; red: corresponding indicator values for all cells. Left: WENO3-MLP; right: WENO3 with KXRCF indicator.

Example 6.2.4 2D Riemann problem Consider the 2D Riemann problem of the equation (31) with the piecewise constant initial condition

$$u_0 = \begin{cases} -1.0, & \text{if } x > 0.5, y > 0.5, \\ -0.2, & \text{if } x < 0.5, y > 0.5, \\ 0.5, & \text{if } x < 0.5, y < 0.5, \\ 0.8, & \text{if } x > 0.5, y < 0.5. \end{cases} \tag{36}$$

For this problem, we are interested in the computational domain $[0, 1]^2$. Following the approach in [1], we solve the equation till the final time $T = 0.5$ on a larger region $[-0.5, 1.5]^2$ with the periodic boundary conditions, so that the numerical solution in the domain $[0, 1]^2$ is unaffected by the boundary conditions. Both the uniform grids and the nonuniform grids with $N = 160$ and $N = 320$ are used. Here we present the contour plots of the numerical solutions on the nonuniform grids in Fig. 11, since the observations on numerical results on the uniform grids are similar and they are omitted to save space. Comparing the numerical solutions in Fig. 11 which are obtained using the WENO3-MLP, the classical WENO3, and the WENO3 with KXRCF indicator, we see that all these three methods yield comparable results. In Fig. 12, we compare the troubled-cells identified in the WENO3-MLP scheme and the WENO3 scheme with KXRCF indicator, on the 320×320 nonuniform grid at different times $t = 0.2$, $t = 0.4$, and $T = 0.5$. Similar to the 1D examples, it is observed that more cells are marked as troubled-cells by the KXRCF indicator than those by the MLP model.

Example 6.2.5 2D KPP problem We consider the 2D KPP problem [24]

$$u_t + (\sin u)_x + (\cos u)_y = 0, \tag{37}$$

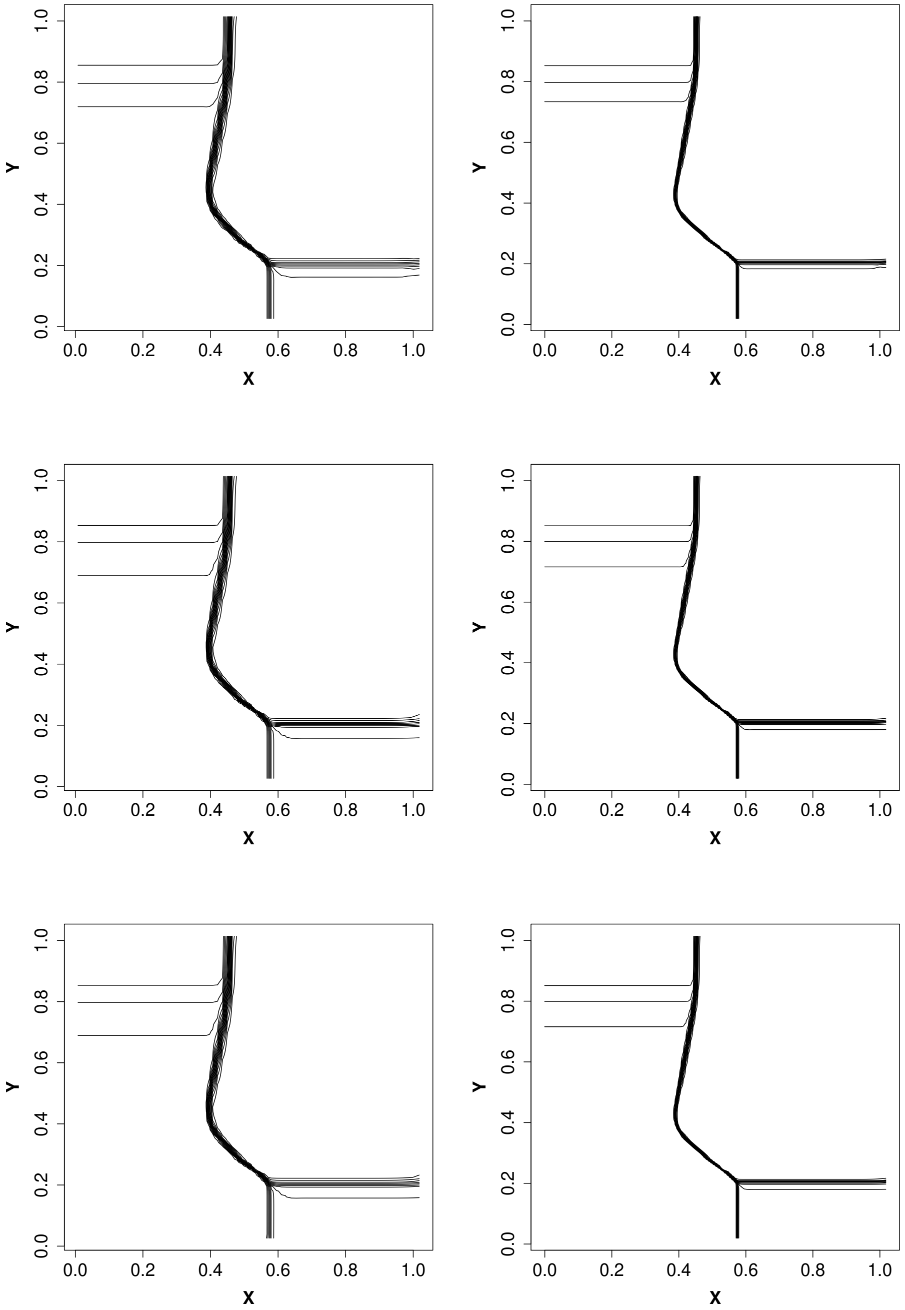

Figure 11: Example 6.2.4, numerical solutions of 2D Riemann problem on nonuniform grid with $c = 0.4$ at $T = 0.5$, 30 equally spaced contour lines from $u = -0.95$ to $u = 0.75$. Top row: WENO3-MLP; middle row: classical WENO3; bottom row: WENO3 with KXRCF indicator. Left column: number of cells $N \times N = 160 \times 160$; right column: number of cells $N \times N = 320 \times 320$.

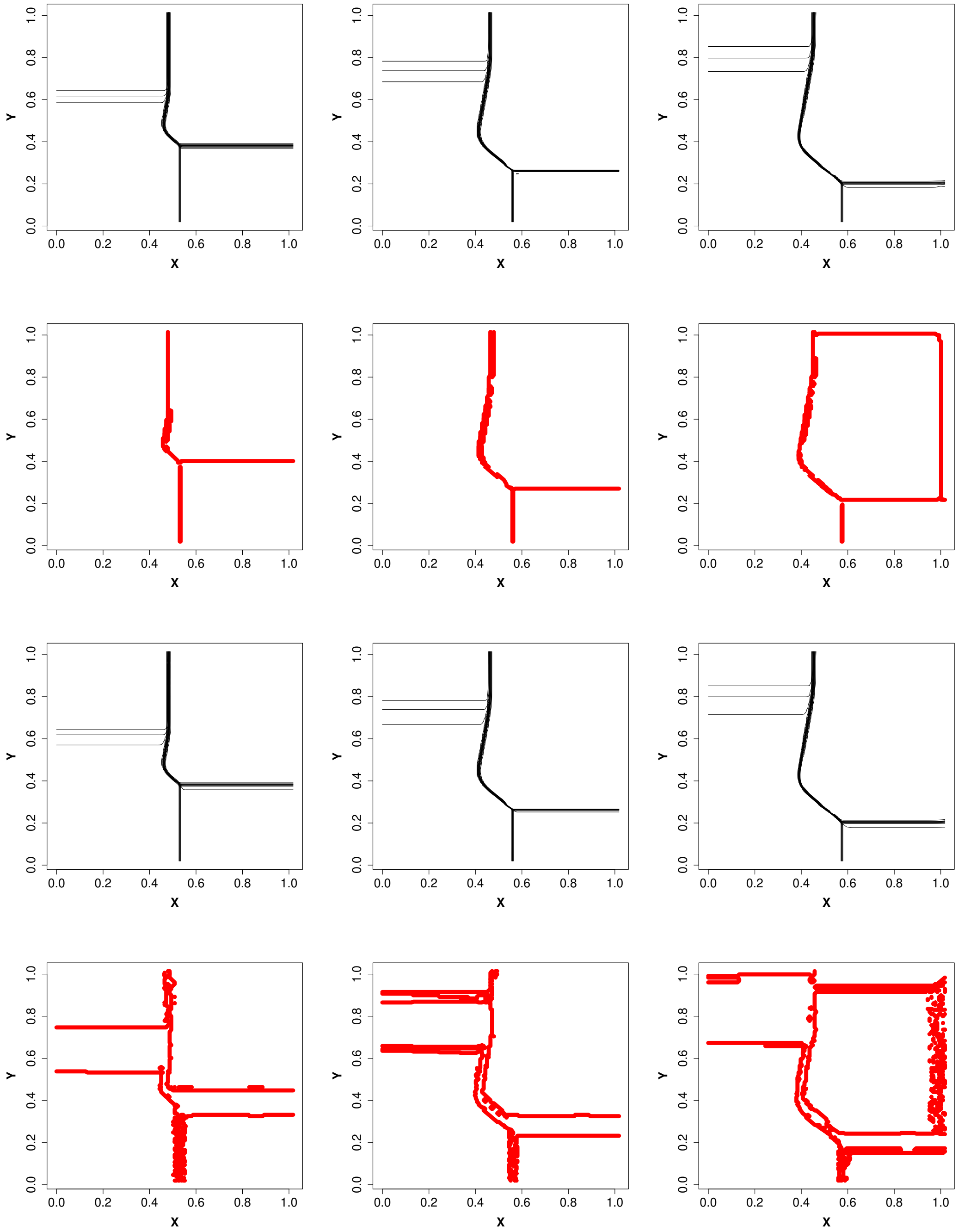

Figure 12: Example 6.2.4, numerical solutions of 2D Riemann problem and troubled-cells (in color red) marked by WENO3-MLP and WENO3 with KXRCF indicator on the nonuniform grid with $c = 0.4$. Number of cells $N \times N = 320 \times 320$. Contour plots of numerical solutions are drawn with 30 equally spaced contour lines from $u = -0.95$ to $u = 0.75$. First row: contour plots for numerical solutions of WENO3-MLP; second row: troubled-cells marked by WENO3-MLP; third row: contour plots for numerical solutions of WENO3 with KXRCF indicator; last row: troubled-cells marked by WENO3 with KXRCF indicator. Left to right: at $t = 0.2$, at $t = 0.4$, at $T = 0.5$.

which has the non-convex flux. The equation is solved on both the uniform grid and the nonuniform grid with the initial condition

$$u_0 = \begin{cases} 14\pi/4, & \text{if } x^2 + y^2 < 1, \\ \pi/4, & \text{otherwise.} \end{cases} \tag{38}$$

The computational domain is $[-2, 2]^2$ with periodic boundary conditions. Since the observations of numerical results are similar on the uniform grids and the nonuniform grids, we only show contour plots of the numerical solutions at $T = 1.0$ on the nonuniform grids with $N = 160$ and $N = 320$ in Fig. 13. Comparable numerical results are obtained for the WENO3-MLP scheme, the classical WENO3 scheme, and the WENO3 scheme with KXRCF indicator. We compare the troubled-cells marked by the WENO3-MLP scheme and the WENO3 scheme with KXRCF indicator, on the 320×320 nonuniform grid at different times $t = 0.4$, $t = 0.8$, and $T = 1.0$. The results are reported in Fig. 14. As the previous examples, the WENO3-MLP scheme marks fewer troubled-cells than the WENO3 with KXRCF indicator does, while comparable numerical results are obtained for both schemes.

Example 6.2.6 Lax problem We solve the Lax problem [25] on the computational domain $[-5, 5]$. Both the uniform grid and the nonuniform grid are used. The problem is the 1D Euler system of equations (32) with the initial condition

$$(\rho_0, u_0, p_0) = \begin{cases} (0.445, 0.698, 3.528) & \text{if } -5 \leq x < 0, \\ (0.5, 0, 0.571) & \text{if } 0 \leq x \leq 5. \end{cases} \tag{39}$$

All of the density, velocity and pressure in the initial condition are discontinuous, and the solution of this problem contains a left-moving rarefaction wave, an intermediate contact discontinuity, and a right-spreading shock wave from discontinuities in the initial condition. The outflow boundary conditions are applied. The exact Riemann solver in [50] is used for computing the reference solution. The problem is solved till the final time $T = 1.3$ on the grids with $N = 100$. In this example, similar to Example 6.2.1, the size of buffer zones is taken as $N_b = N_{b,\kappa} = 7$ for the WENO3-MLP and the WENO3 with KXRCF indicator, which is increased from the default value in the other examples for better stability of the simulation. Fig. 15 reports the numerical solutions of three schemes for the density ρ on both the whole computational domain and an enlarged portion near the contact discontinuity. It shows that comparable numerical results are obtained for these three different numerical methods, and verifies the nonlinear stability of the proposed WENO3-MLP hybrid finite volume scheme. Fig. 16 reports the comparison of troubled-cells marked by the WENO3-MLP scheme and the WENO3 scheme with KXRCF indicator on the nonuniform grid. Again, we observe that more cells are marked as troubled-cells by the WENO3 scheme with KXRCF indicator than those by the WENO3-MLP scheme.

Example 6.2.7 Double Mach reflection We solve the "double Mach reflection" problem [6]. Consider the 2D Euler system of equations (33) on the computational domain $[0, 4] \times [0, 1]$. A reflecting wall lies at the bottom of the computational domain

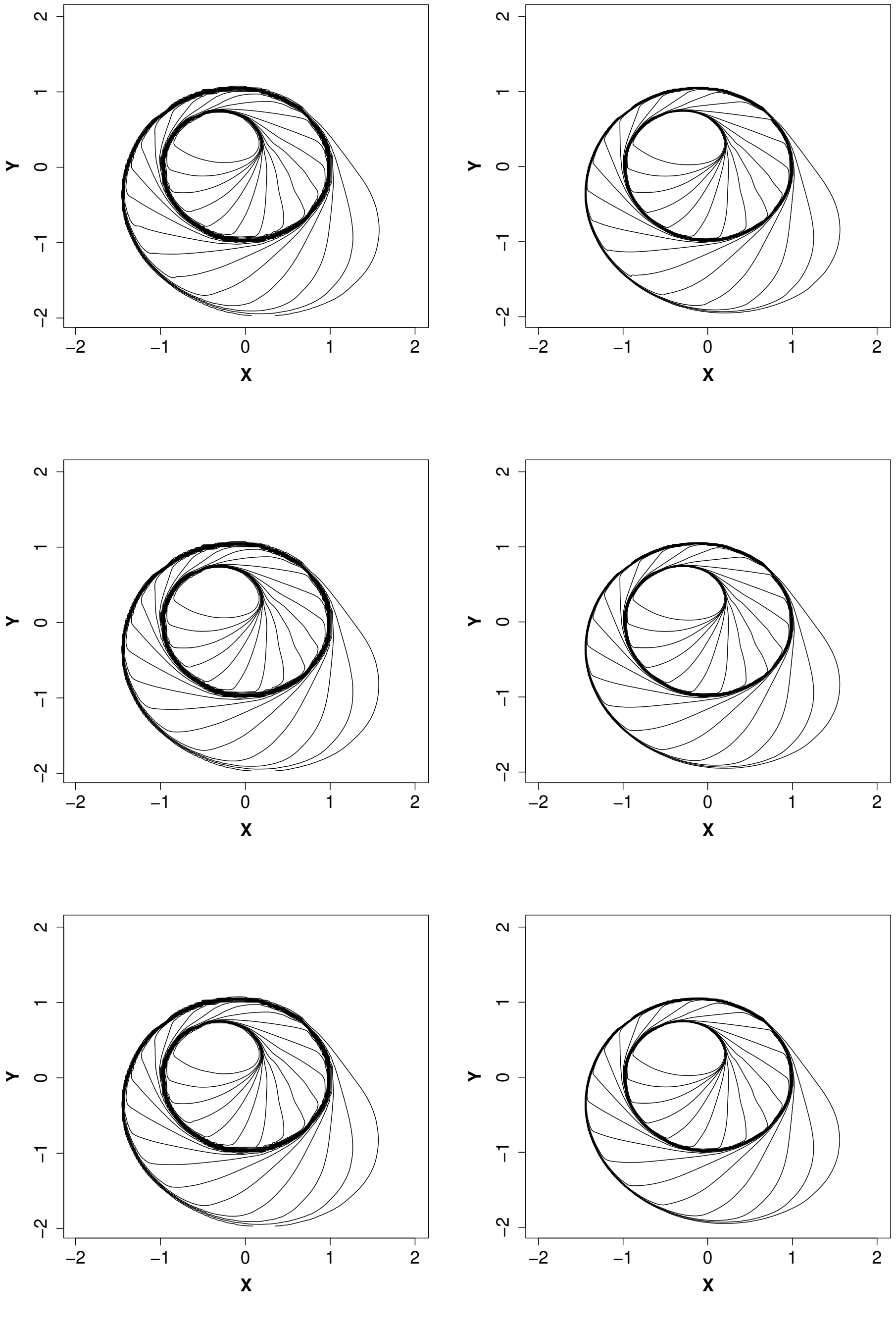

Figure 13: Example 6.2.5, numerical solutions of the 2D KPP problem on the nonuniform grids with $c = 0.4$ at $T = 1.0$, 30 equally spaced contour lines from $u = 1.0$ to $u = 10.8$. Top row: WENO3-MLP; middle row: classical WENO3; bottom row: WENO3 with KXRCF indicator. Left column: number of cells $N \times N = 160 \times 160$; right column: number of cells $N \times N = 320 \times 320$.

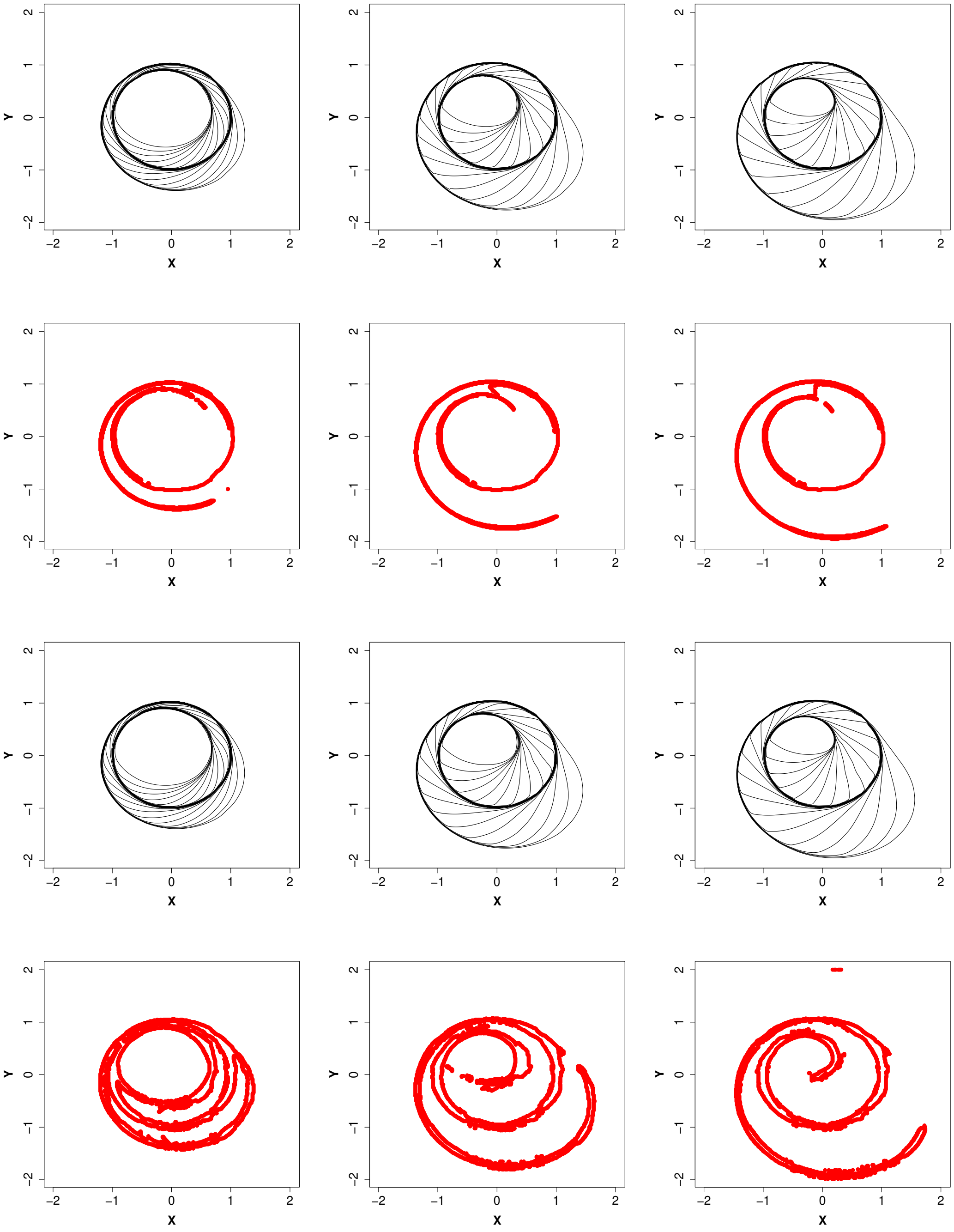

Figure 14: Example 6.2.5, numerical solutions of 2D KPP problem and troubled-cells (in color red) marked by WENO3-MLP and WENO3 with KXRCF indicator on the nonuniform grid with $c = 0.4$. Number of cells $N \times N = 320 \times 320$. Contour plots are drawn with 30 equally spaced contour lines from $u = 1.0$ to $u = 10.8$. First row: contour plots for WENO3-MLP; second row: troubled-cells marked by WENO3-MLP; third row: contour plots for WENO3 with KXRCF indicator; last row: troubled-cells marked by WENO3 with KXRCF indicator. Left to right: at $t = 0.4$, at $t = 0.8$, at $T = 1.0$.

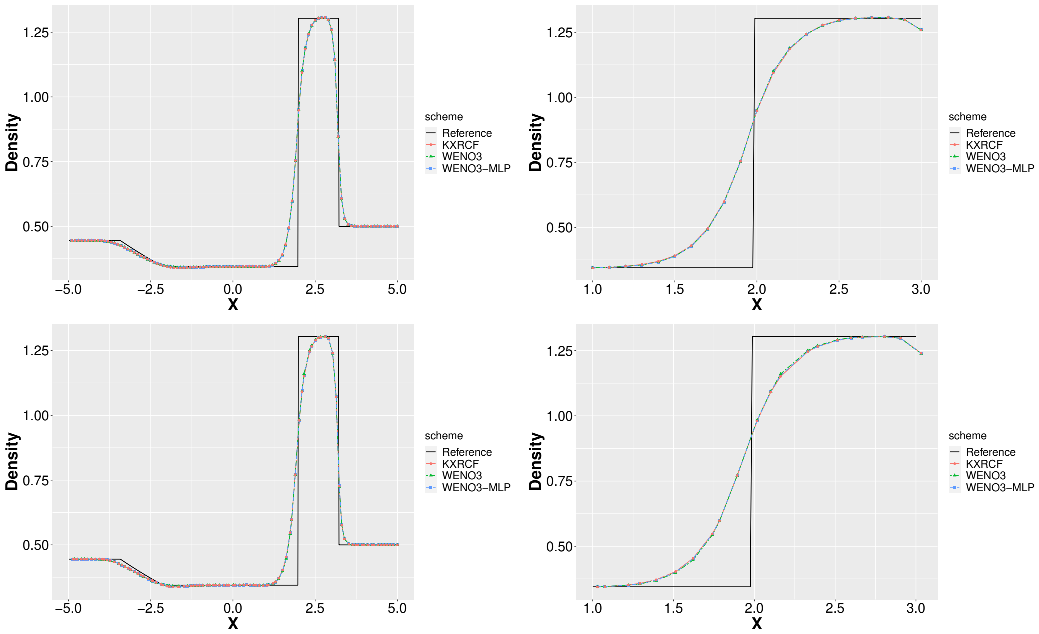

Figure 15: Example 6.2.6, numerical solutions of the Lax problem for the density at $T = 1.3$. Number of computational cells $N = 100$. Top row: uniform grid; bottom row: nonuniform grid with $c = 0.4$. Left column: the whole computational domain; right column: an enlarged portion near the contact discontinuity.

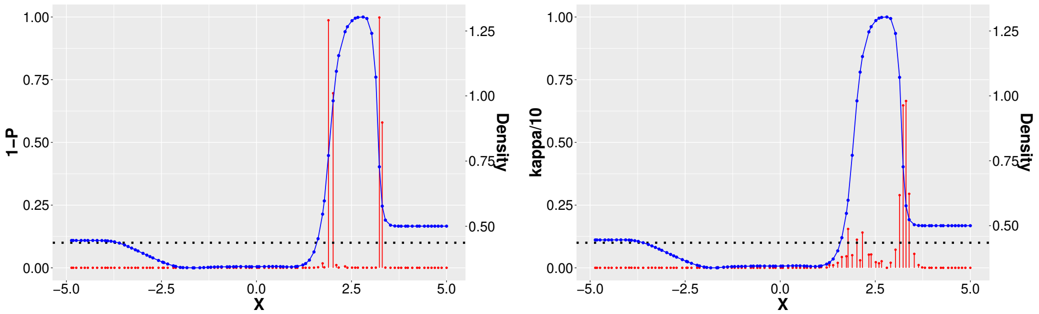

Figure 16: Example 6.2.6, numerical solutions of the density for the Lax problem and troubled-cells (i.e., the indicator values are above the dashed cutoff line) marked by WENO3-MLP and WENO3 with KXRCF indicator on the nonuniform grid with $c = 0.4$ at $T = 1.3$. Number of computational cells $N = 100$. Blue: numerical solutions; red: corresponding indicator values for all cells. Left: WENO3-MLP; right: WENO3 with KXRCF indicator.

for $\frac{1}{6} \leq x \leq 4$. Initially a right-moving Mach 10 shock is positioned at $x = \frac{1}{6}, y = 0$, and makes a 60° angle with the x-axis. At the bottom of the computational domain, the reflective boundary condition is used from $x = \frac{1}{6}$ to $x = 4$, and the exact post-shock condition is imposed from $x = 0$ to $x = \frac{1}{6}$. The top boundary is set to describe the exact motion of the Mach 10 shock. Prescribed exact post-shock inflow and free outflow boundary conditions are used for the left and right boundaries respectively. The initial condition of the double Mach reflection problem is

$$(\rho_0, u_0, v_0, p_0) = \begin{cases} (8, 8.25\cos(\pi/6), -8.25\sin(\pi/6), 116.5) & \text{if } x < \frac{1}{6} + \frac{y}{\sqrt{3}}, \\ (1.4, 0, 0, 1) & \text{if } x \geq \frac{1}{6} + \frac{y}{\sqrt{3}}. \end{cases} \tag{40}$$

We solve the problem till the final time $T = 0.2$ on both the uniform grid and the nonuniform grid with $N_x \times N_y = 960 \times 240$ computational cells. Since the observations of numerical results are similar on the uniform grid and the nonuniform grid, we only show the results on the nonuniform grid to save spaces. Fig. 17 shows the contour plots of numerical solutions on the nonuniform grid with $c = 0.4$ for the density in the domain $[0, 3] \times [0, 1]$. Comparable numerical results are observed for these three different numerical schemes. We also compare troubled-cells identified by the WENO3-MLP scheme and the WENO3 scheme with KXRCF indicator on the nonuniform grid, and the results are reported in Fig. 18. It is observed that more cells are marked as troubled-cells by the KXRCF indicator, which is consistent with the results of the other examples in this paper.

Example 6.2.8 Rayleigh-Taylor instability Rayleigh-Taylor instability happens on an interface between fluids with different densities when an acceleration is directed from the heavy fluid to the light fluid. The instability has a fingering nature, with bubbles of light fluid rising into the ambient heavy fluid and spikes of heavy fluid falling into the light fluid [12, 46, 60, 62]. This is a standard benchmark problem to test the resolution of high-order schemes. More small structures in the numerical solutions, which are driven by the numerical viscosity of the scheme, indicate smaller numerical viscosity and higher resolution of the scheme. We solve the problem on the computational domain $[0, \frac{1}{4}] \times [0, 1]$. Initially the interface is at $y = \frac{1}{2}$. The heavy fluid with density $\rho = 2$ is below the interface, and the light fluid with density $\rho = 1$ is above the interface with the acceleration in the positive y-direction. The pressure p is continuous across the interface, and a small perturbation is given to the y-direction fluid speed. The initial condition of the Rayleigh–Taylor instability problem is formulated as

$$(\rho_0, u_0, v_0, p_0) = \begin{cases} (2, 0, -0.025c' \cdot \cos(8\pi x), 1 + 2y) & \text{if } 0 \leq y < \dfrac{1}{2}, \\ (1, 0, -0.025c' \cdot \cos(8\pi x), \dfrac{3}{2} + y) & \text{if } \dfrac{1}{2} \leq y \leq 1, \end{cases} \tag{41}$$

where the speed of sound $c' = \sqrt{\frac{\gamma p}{\rho}}$ and the ratio of specific heat $\gamma = \frac{5}{3}$. Reflective boundary conditions are imposed for the left and right boundaries. At the top boundary, the flow values are set as $(\rho, u, v, p) = (1, 0, 0, 2.5)$, and at the bottom boundary, the flow values are set as $(\rho, u, v, p) = (2, 0, 0, 1)$. We solve the 2D Euler system of equations (33) with the source terms ρ added the right hand side of the third equation

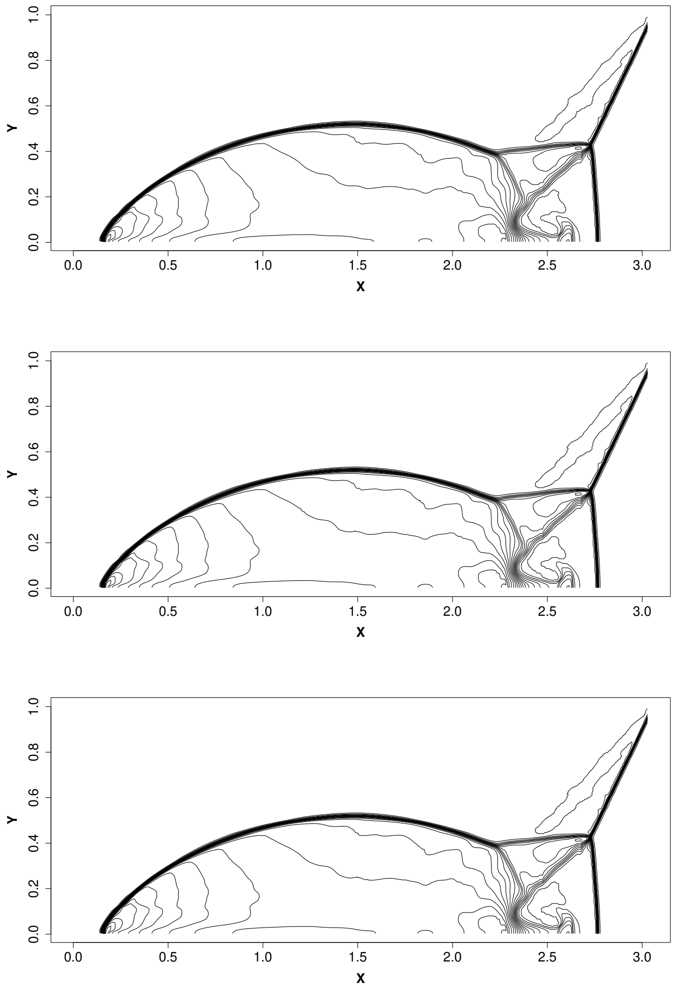

Figure 17: Example 6.2.7, density contours of the numerical solutions for the double Mach reflection problem on the nonuniform grid with $c = 0.4$, in the domain $[0, 3] \times [0, 1]$ at $T = 0.2$. 30 equally spaced contour lines from $\rho = 1.5$ to $\rho = 21.0$. Number of computational cells $N_x \times N_y = 960 \times 240$. Top: WENO3-MLP ; middle: classical WENO3; bottom: WENO3 with KXRCF indicator.

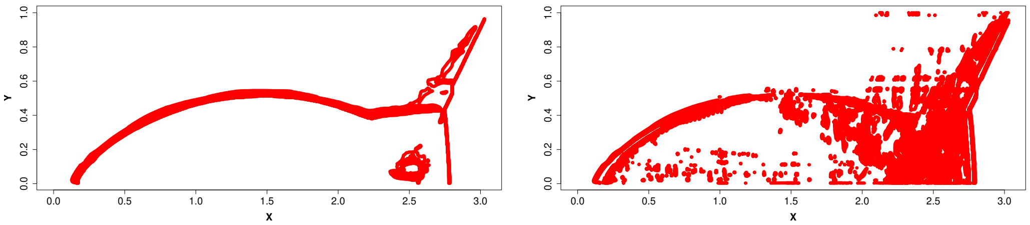

Figure 18: Example 6.2.7, troubled-cells (in color red) marked by the WENO3-MLP scheme and the WENO3 scheme with KXRCF indicator for the double Mach reflection problem on the nonuniform grid with $c = 0.4$, in the domain $[0, 3] \times [0, 1]$ at $T = 0.2$. Number of computational cells $N_x \times N_y = 960 \times 240$. Left: troubled-cells marked by WENO3-MLP; right: troubled-cells marked by WENO3 with KXRCF indicator.

and ρv added to the right hand side of the fourth equation. The simulations are run till the final time $T = 1.95$ on both the uniform grid and the nonuniform grid with $N_x \times N_y = 240 \times 960$ computational cells. Numerical results are reported in Fig. 19. The top row of Fig. 19 shows density contours with 30 equally spaced contour lines using these three different numerical schemes on the uniform grid. We observe more small structures in the numerical solution of the WENO3-MLP scheme, whereas the WENO3 scheme with KXRCF indicator produces similar numerical solution to that of the classical WENO3 scheme. Troubled-cells identified by the WENO3-MLP scheme and the WENO3 scheme with KXRCF indicator on the uniform grid are shown in the left two pictures of Fig. 20. Similar to the previous examples, more cells are marked as troubled-cells by the WENO3 scheme with KXRCF indicator. On the nonuniform grid with $c = 0.4$, we have consistent observations with those on the uniform grid, as shown at the bottom row of Fig. 19. Comparable numerical solutions are obtained for the classical WENO3 scheme and the WENO3 scheme with KXRCF indicator, while WENO3-MLP scheme generates more small structures and shows a better resolution. Troubled-cells identified by the WENO3-MLP scheme and the WENO3 scheme with KXRCF indicator on the nonuniform grid are shown in these two pictures of the right side in Fig. 20. Less cells are identified as troubled-cells by the WENO3-MLP scheme, than those marked by the WENO3 scheme with KXRCF indicator. Hence the linear scheme is used on a larger region in the WENO3-MLP hybrid scheme than that in the other hybrid scheme (the WENO3 scheme with KXRCF indicator), which leads to smaller numerical viscosity and higher resolution.

7 Concluding remarks

Motivated by recent development in applying ANN based troubled-cell indicators to designing hybrid finite difference WENO schemes, in this paper we develop a third-order hybrid finite volume WENO method with a MLP troubled-cell indicator for solving hyperbolic conservation laws. The method is designed on general Cartesian

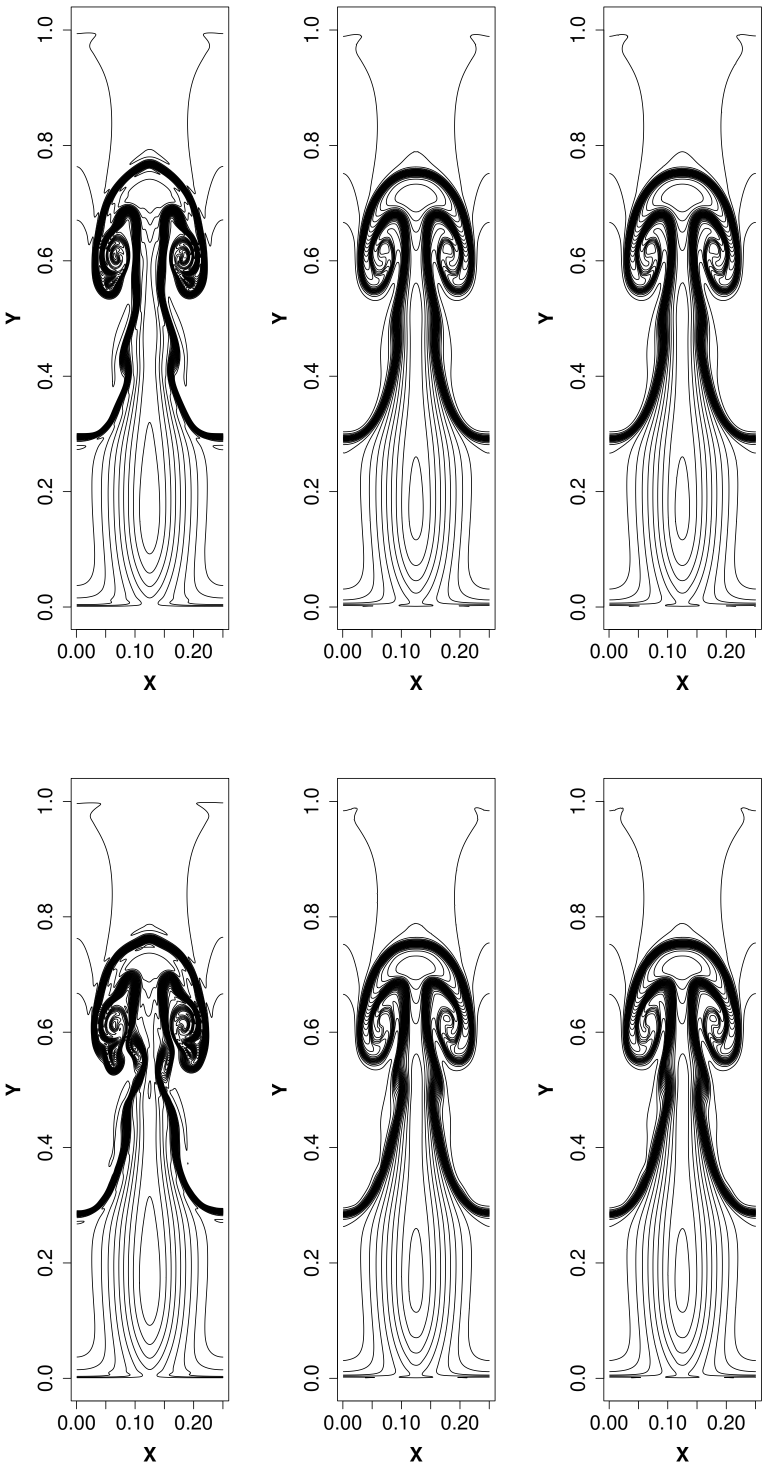

Figure 19: Example 6.2.8, density contours of the numerical solutions for the Rayleigh-Taylor instability problem on the uniform grid and the nonuniform grid with $c = 0.4$ at $T = 1.95$, 30 equally spaced contour lines from $\rho = 0.9$ to $\rho = 2.35$. Number of computational cells $N_x \times N_y = 240 \times 960$. Top row: on uniform grid; bottom row: on nonuniform grid with $c = 0.4$. Left column: WENO3-MLP; middle column: classical WENO3; right column: WENO3 with KXRCF indicator.

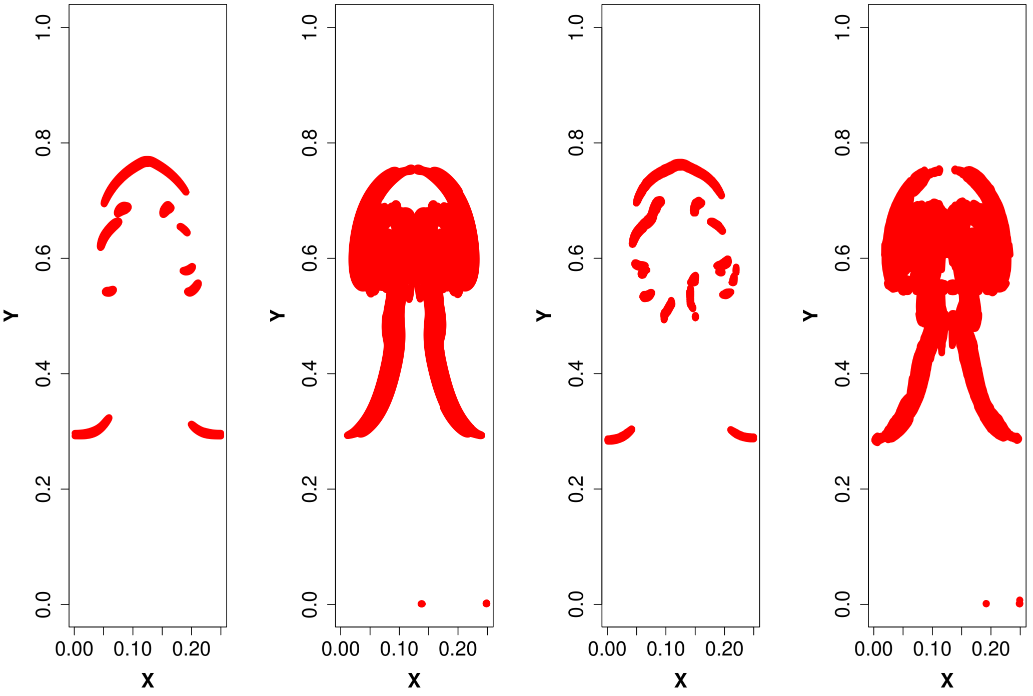

Figure 20: Example 6.2.8, troubled-cells (in color red) marked by the WENO3-MLP scheme and the WENO3 scheme with KXRCF indicator for the Rayleigh-Taylor instability problem on the uniform grid and the nonuniform grid with $c = 0.4$ at $T = 1.95$. Number of computational cells $N_x \times N_y = 240 \times 960$. Pictures from left to right: troubled-cells marked by WENO3-MLP on uniform grid; troubled-cells marked by WENO3 with KXRCF indicator on uniform grid; troubled-cells marked by WENO3-MLP on nonuniform grid with $c = 0.4$; troubled-cells marked by WENO3 with KXRCF indicator on nonuniform grid with $c = 0.4$.

meshes including arbitrary nonuniform grids, without the grid-restriction of finite difference WENO schemes. Extensive numerical examples, including scalar equations with both convex and non-convex fluxes and the Euler systems of equations on uniform and nonuniform grids of 1D and 2D domains, are solved by the proposed new method to show its desired third-order accuracy and nonlinear stability. Numerical experiments verify significant accuracy improvement and computational-cost saving of the proposed new scheme over the original third-order finite volume WENO scheme, which is consistent with the property of the existing hybrid WENO methods in the literature. Moreover, comparisons of the MLP troubled-cell indicator developed in this paper with the widely-used KXRCF troubled-cell indicator show that the MLP troubled-cell indicator accurately identifies the troubled-cells and marks less troubled-cells for problems with nonsmooth solutions, which leads to a larger computational domain covered by the linear scheme, hence more efficient computations for the hybrid finite volume WENO scheme. It is also verified that the MLP troubled-cell indicator performs very well on arbitrary nonuniform grids, although it is trained offline on uniform grids.

We focus on the third-order scheme in this paper to explore the methodology of developing hybrid finite volume WENO method on general Cartesian meshes with an ANN based troubled-cell indicator. However, the approach proposed here is expected to have the potential to be further extended to higher-order schemes and unstructured triangular meshes. In the numerical experiments, the WENO3-MLP scheme shows comparable resolution to the other two schemes in some examples, and much higher resolution in solving the typical benchmark problem of testing resolution for high-order schemes [46, 62], the Rayleigh-Taylor instability problem. Note that there have been extensive efforts in the literature to achieve better resolution for high-order WENO schemes, for example, by designing new smoothness indicators, new nonlinear weights, more effective troubled-cell indicators for hybrid schemes, etc. Although it is not the goal of this paper to design a hybrid WENO scheme with a better resolution than the existing methods in the literature, they will be very interesting and important topics to improve the hybrid finite volume WENO scheme proposed in this paper for higher-order accuracy and higher resolution, and to compare the new hybrid methods more comprehensively with other hybrid schemes with different troubled-cell indicators. These future work will be carried out in our next research.

Data Availability

Data sets generated during the current study of this paper can be requested by contacting the authors of the paper.

Declarations

Conflict of interest

The authors declare that we have no conflict of interest.

References

[1] Arbogast, T., Huang, CS. Kuo, MH.: RBF WENO reconstructions with adaptive order and applications to conservation laws. J. Sci. Comput. 91, 51 (2022). https://doi.org/10.1007/s10915-022-01827-6

[2] Balsara, D.S., Garain, S., Florinski, V., Boscheri, W.: An efficient class of WENO schemes with adaptive order for unstructured meshes, J. Comput. Phys. 404, 109062 (2020)

[3] Balsara, D.S., Garain, S., Shu, C.-W.: An efficient class of WENO schemes with adaptive order. J. Comput. Phys. 326, 780–804 (2016)

[4] Bezgin, D., Schmidt, S., Adams, N.: WENO3-NN: A maximum-order three-point data-driven weighted essentially non-oscillatory scheme. J. Comput. Phys. 452 (2022). https://doi.org/10.1016/j.jcp.2021.110920

[5] Borges, R., Carmona, M., Costa, B., Don, W.S.: An improved weighted essentially non-oscillatory scheme for hyperbolic conservation laws. J. Comput. Phys. 227, 3191–3211 (2008)

[6] Colella, P., Woodward, P.: The piecewise parabolic method (PPM) for gas dynamical simulations. J. Comput. Phys. 54, 174-201 (1984)

[7] Costa, B., Don, W.S.: Multi-domain hybrid spectral-WENO methods for hyperbolic conservation laws. J. Comput. Phys. 224 (2), 970-991 (2007)

[8] Costa, B., Don, W.S., Gottlieb, D., Sendersky, R.: Two-dimensional multi-domain hybrid spectral-WENO methods for conservation laws. Commun. Comput. Phys. 1 (3), 548-574 (2006)

[9] Cox, D.R.: The Regression Analysis of Binary Sequences. J. R. Stat. Soc. Series B Stat. Methodol. 20 (2), 215-232 (1958)

[10] Cybenko, G.: Approximation by superpositions of a sigmoidal function. Math. Control Signal Syst. 2, 303–314 (1989). https://doi.org/10.1007/BF02551274

[11] Dumbser, M., Kaser, M., Titarev, V.A., Toro, E.F.: Quadrature-free non-oscillatory finite volume schemes on unstructured meshes for nonlinear hyperbolic systems. J. Comput. Phys. 226, 204-243 (2007)

[12] Gardner, C.L., Glimm, J., McBryan, O., Menikoff, R., Sharp, D.H., Zhang, Q.: The dynamics of bubble growth for Rayleigh–Taylor unstable interfaces. Phys. Fluids 31, 447–465 (1988)

[13] Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, PMLR 9:249-256 (2010)

[14] Haykin, S.: Neural networks and learning machines. Pearson Education, Upper Saddle River, NJ, Third edition (2009)

[15] Henrick, A., Aslam, T., Powers, J.: Mapped weighted essentially non-oscillatory schemes: achieving optimal order near critical points. J. Comput. Phys. 207, 542-567 (2005)

[16] Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neural Netw. 2 (5), 359-366 (1989)

[17] Hu, C., Shu, C.-W.: Weighted essentially non-oscillatory schemes on triangular meshes. J. Comput. Phys. 150, 97–127 (1999)

[18] Hu, R., Zhang, Y.-T.: High order absolutely convergent fast sweeping methods with multi-resolution WENO local solvers for Eikonal and factored Eikonal equations. J. Sci. Comput. 99, 67 (2024)

[19] Huang, CS., Arbogast, T., Tian, C.: Multidimensional WENO-AO reconstructions using a simplified smoothness indicator and applications to conservation laws. J. Sci. Comput. 97, 8 (2023). https://doi.org/10.1007/s10915-023-02319-x

[20] Jiang, G.S., Shu, C.-W.: Efficient implementation of weighted ENO schemes. J. Comput. Phys. 126, 202–228 (1996)

[21] Jiang, T., Zhang, Y.-T.: Krylov single-step implicit integration factor WENO methods for advection–diffusion–reaction equations. J. Comput. Phys. 311, 22-44 (2016)

[22] Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. (2014). arXiv:1412.6980

[23] Krivodonova, L., Xin, J., Remacle, J.-F., Chevaugeon, N., Flaherty, J.E.: Shock detection and limiting with discontinuous Galerkin methods for hyperbolic conservation laws. Appl. Numer. Math. 48, 323-338 (2004)

[24] Kurganov, A., Petrova, G., Popov, B.: Adaptive semidiscrete central-upwind schemes for nonconvex hyperbolic conservation laws. SIAM J. Sci. Comput. 29 (6), 2381-2401 (2007). https://doi.org/10.1137/040614189

[25] Lax, P.D. Weak solutions of nonlinear hyperbolic equations and their numerical computation. Commun. Pure Appl. Math. 7, 159–193 (1954)

[26] LeCun, Y., Boser, B., Denker, J. et al.: Handwritten digit recognition with a back-propagation network. Advances in Neural Information Processing Systems 2 (NeurIPS), D. Touretzky, Ed., Denver, CO, USA (1990)

[27] LeCun, Y., Bottou, L., Bengio Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324 (1998). doi: 10.1109/5.726791

[28] Lee, K., Choi, J-I., Yoon, J. Improving the third-order WENO schemes by using exponential polynomial space with a locally optimized shape parameter. Comput. Math. Appl. 149, 24-37 (2023)

[29] Levy, D., Puppo, G., Russo, G.: Compact central WENO schemes for multidimensional conservation laws. SIAM J. Sci. Comput. 22, 656-672 (2000)

[30] Li, G., Lu, C., Qiu, J.: Hybrid well-balanced WENO schemes with different indicators for shallow water equations. J. Sci. Comput. 51, 527–559 (2012)

[31] Li, G., Qiu, J.: Hybrid weighted essentially non-oscillatory schemes with different indicators. J. Comput. Phys. 229, 8105-8129 (2010)

[32] Li, L., Zhu, J., Zhang, Y.-T.: Absolutely convergent fixed-point fast sweeping WENO methods for steady state of hyperbolic conservation laws. J. Comput. Phys. 443, 110516 (2021)

[33] Liu, X.-D., Osher, S., Chan, T.: Weighted essentially non-oscillatory schemes. J. Comput. Phys. 115, 200–212 (1994)

[34] Liu, Y., Zhang, Y.-T.: A robust reconstruction for unstructured WENO schemes. J. Sci. Comput. 54, 603–621 (2013)

[35] Lu, D., Zhang, Y.-T.: Computational complexity study on Krylov integration factor WENO method for high spatial dimension convection-diffusion problems. J. Sci. Comput. 73, 980-1027 (2017)

[36] Maas, A.L., Hannun, A.Y., Ng, A.Y.: Rectifier nonlinearities improve neural network acoustic models. Proceedings of International Conference on Machine Learning, 30 (1), 3 (2013)

[37] Miksis, Z.M., Zhang, Y.-T.: Sparse-grid implementation of fixed-point fast sweeping WENO schemes for Eikonal equations. Commun. Appl. Math. Comput. 6, 3-29 (2024)

[38] Paszke, A. et al.: PyTorch: an imperative style, high-performance deep learning library. NeurIPS, 32, 8026–8037 (2019)

[39] Pirozzoli, S.: Conservative hybrid compact-WENO schemes for shock-turbulence interaction. J. Comput. Phys. 178, 81-117 (2002)

[40] Qiu, J., Shu, C.-W.: A comparison of troubled-cell indicators for runge-kutta discontinuous Galerkin methods using weighted essentially nonoscillatory limiters. SIAM J. Sci. Comput. 27 (3): 995-1013 (2005)

[41] Ray, D., Hesthaven, J.S.: An artificial neural network as a troubled-cell indicator. J. Comput. Phys. 367 (2018)

[42] Ray, D., Hesthaven, J.S.: Detecting troubled-cells on two-dimensional unstructured grids using a neural network. J. Comput. Phys. 397 (2019)

[43] Rosenblatt, F.: The perceptron: a probabilistic model for information storage and organization in the brain. Psychol. Rev. 65, 386–408 (1958)

[44] Rumelhart, D., Hinton, G., Williams, R.: Learning representations by back-propagating errors. Nature 323, 533–536 (1986)

[45] Shi, J., Hu, C., Shu, C.-W.: A technique of treating negative weights in WENO schemes. J. Comput. Phys. 175, 108–127 (2002)

[46] Shi, J., Zhang, Y.-T., Shu, C.-W.: Resolution of high order WENO schemes for complicated flow structures. J. Comput. Phys. 186 (2), 690–696 (2003)

[47] Shu, C.-W.: Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. In: Quarteroni, A. (eds) Advanced Numerical Approximation of Nonlinear Hyperbolic Equations. Lecture Notes in Mathematics, vol. 1697. Springer, Berlin, Heidelberg (1998)

[48] Shu, C.-W., Osher, S.: Efficient implementation of essentially non-oscillatory shock-capturing schemes. J. Comput. Phys. 77, 439–471 (1988)

[49] Sun, Z., Wang, S., Chang, LB., Xing, Y., Xiu, D.: Convolution neural network shock detector for numerical solution of conservation laws. Commun. Comput. Phys. 28 (5). 2075-2108 (2020). https://doi:10.4208/cicp.OA-2020-0199

[50] Toro, E. F. Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction, 3rd edition. Springer, Berlin, Heidelberg (2009)

[51] Tsoutsanis, P., Adebayo, E.M., Merino, A.C., Arjona, A.P., Skote, M.: CWENO finite-volume interface capturing schemes for multicomponent flows using unstructured meshes. J. Sci. Comput. 89 (3), 1-27 (2021)

[52] Tsybulnik, E., Zhu, X., Zhang, Y.-T.: Efficient sparse-grid implementation of a fifth-order multi-resolution WENO scheme for hyperbolic equations. Commun. Appl. Math. Comput. 5, 1339-1364 (2023)

[53] Veiga, M.H., Abgrall, R.: Neural network based limiter with transfer learning, Commun. Appl. Math. Comput. 5 (2), 532-572 (2023). https://doi.org/10.1007/s42967-020-00087-1.

[54] Vuik, M.J., Ryan, J.K.: Automated parameters for troubled-cell indicators using outlier detection. SIAM J. Sci. Comput. 38, A84-A104 (2016)

[55] Wang, B.S., Don, W.S., Gao, Z., Wang, Y.H., Wen, X.: Hybrid compact-WENO finite difference scheme with radial basis function based shock detection method for hyperbolic conservation laws. SIAM J. Sci. Comput. 40 (6), A3699-A3714 (2018)

[56] Wen, X., Don, W.S., Gao, Z., Hesthaven, J.S.: An edge detector based on artificial neural network with application to hybrid compact WENO finite difference scheme. J. Sci. Comput. 83, 49 (2020)

[57] Wu, L., Zhang, Y.-T., Zhang, S., Shu, C.-W.: High order fixed-point sweeping WENO methods for steady state of hyperbolic conservation laws and its convergence study. Commun. Comput. Phys. 20, 835-869 (2016)

[58] Xu, Z., Zhang, Y.-T.: High-order exponential time differencing multi-resolution alternative finite difference WENO methods for nonlinear degenerate parabolic equations. arXiv preprint arXiv:2406.05237. (2024)

[59] Xue, Z., Xia, Y., Li, C., Yuan, X.: A simplified multilayer perceptron detector for the hybrid WENO scheme. Comput. Fluids 244, Article 105584 (2022)

[60] Young, Y.-N., Tufo, H., Dubey, A., Rosner, R.: On the miscible Rayleigh–Taylor instability: two and three dimensions. J. Fluid Mech. 447, 377-408 (2001)

[61] Yu, X., Shu, C.-W.: Multi-layer perceptron estimator for the total variation bounded constant in limiters for discontinuous Galerkin methods. La Matematica 1, 53–84 (2022). https://doi.org/10.1007/s44007-021-00004-9

[62] Zhang, Y.-T., Shi, J., Shu, C.-W., Zhou, Y.: Numerical viscosity and resolution of high-order weighted essentially nonoscillatory schemes for compressible flows with high Reynolds numbers. Phys. Rev. E 68 (4), 046709 (2003)

[63] Zhang, Y.-T., Shu, C.-W.: High order WENO schemes for Hamilton-Jacobi equations on triangular meshes. SIAM J. Sci. Comput. 24, 1005–1030 (2003)

[64] Zhang, Y.-T., Shu, C.-W.: Third order WENO scheme on three dimensional tetrahedral meshes. Commun. Comput. Phys. 5, 836-848 (2009)

[65] Zhao, Z., Qiu, J.: A Hermite WENO scheme with artificial linear weights for hyperbolic conservation laws. J. Comput. Phys. 417, 109583 (2020)

[66] Zhu, J., Qiu, J.: A Finite Difference Mapped WENO Scheme with Unequal-Size Stencils for Hyperbolic Conservation Laws. J. Sci. Comput. 93 (3), 72 (2022)

[67] Zhu, J., Shu, C.-W.: A new type of multi-resolution WENO schemes with increasingly higher order of accuracy on triangular meshes. J. Comput. Phys. 392, 19–33 (2019)

[68] Zhu, X., Zhang, Y.-T.: Fast sparse grid simulations of fifth order WENO scheme for high dimensional hyperbolic PDEs. J. Sci. Comput. 87 (2), 44 (2021)